Treasury Board of Canada Secretariat
Secrétariat du Conseil du Trésor du Canada
Canada

Government of Canada Guideline on Vulnerability Management

Published: 2025-11-17

Published by Treasury Board of Canada, Secretariat
90 Elgin, Ottawa, Ontario, K1A 0R5, Canada

Catalogue Number: BT22-293/2025E-PDF
ISBN: 978-0-660-97557-3

This document is available on the Government of Canada website, Canada.ca

This document is available in alternative formats upon request.

Aussi offert en français sous le titre : Lignes directrices sur la gestion des vulnérabilités du gouvernement du Canada

Government of Canada Guideline on Vulnerability Management

On this page

1. Introduction

▼ In this section

1.1 Background

The Government of Canada's Policy on Government Security and Appendix B: Mandatory Procedures for Information Technology Security Control of the Directive on Security Management mandate departments to "implement measures to protect information systems, their components and the information they process and transmit against attacks that leverage vulnerabilities in information systems to affect their integrity and that could have an impact on their availability or confidentiality (for example, malicious code)." [1] This includes implementing corrective actions such as applying patches to address vulnerabilities. In addition, the Canadian Centre for Cyber Security (Cyber Centre) prioritizes patching operating systems and applications as the second most important information technology (IT) security action an organization can undertake to minimize intrusions and their impacts. [2]

This document incorporates foundational elements of the SANS Institute's Vulnerability Management Maturity Model as well as the National Institute of Standards and Technology NIST SP 800-40 Rev. 4: Guide to Enterprise Patch Management Planning: Preventive Maintenance for Technology) for enterprise patch management technologies. It provides specific guidance on developing, maintaining and managing a vulnerability and patch management program.

Vulnerability management (VM) is fundamentally a risk management function designed to identify, assess and mitigate information security risks associated with information technology (IT) weaknesses. By focusing on remediating or reducing risk to acceptable levels, VM enables an organization to continue delivering its mission effectively and securely. VM serves to reduce the potential for exploitation in a way that involves considerably less time and financial resources than reacting to an incident after an exploitation has occurred. Thus, a well-executed VM program aims to optimize risk management and resource allocation.

1.2 Purpose and scope

The purpose of this document is to provide guidance on developing, maintaining and managing a VM program within a Government of Canada (GC) department or agency.

This guidance aligns with the following GC documents and instruments:

- Policy on Government Security
- Policy on Service and Digital
- Directive on Security Management
- Government of Canada's Enterprise Cyber Security Strategy
- Government of Canada Cyber Security Event Management Plan (GC CSEMP)
- Patch Management Guidance
- Security Vulnerabilities and Patches Explained: IT Security Bulletin for the Government of Canada (ITSB-96)

1.3 Target audience

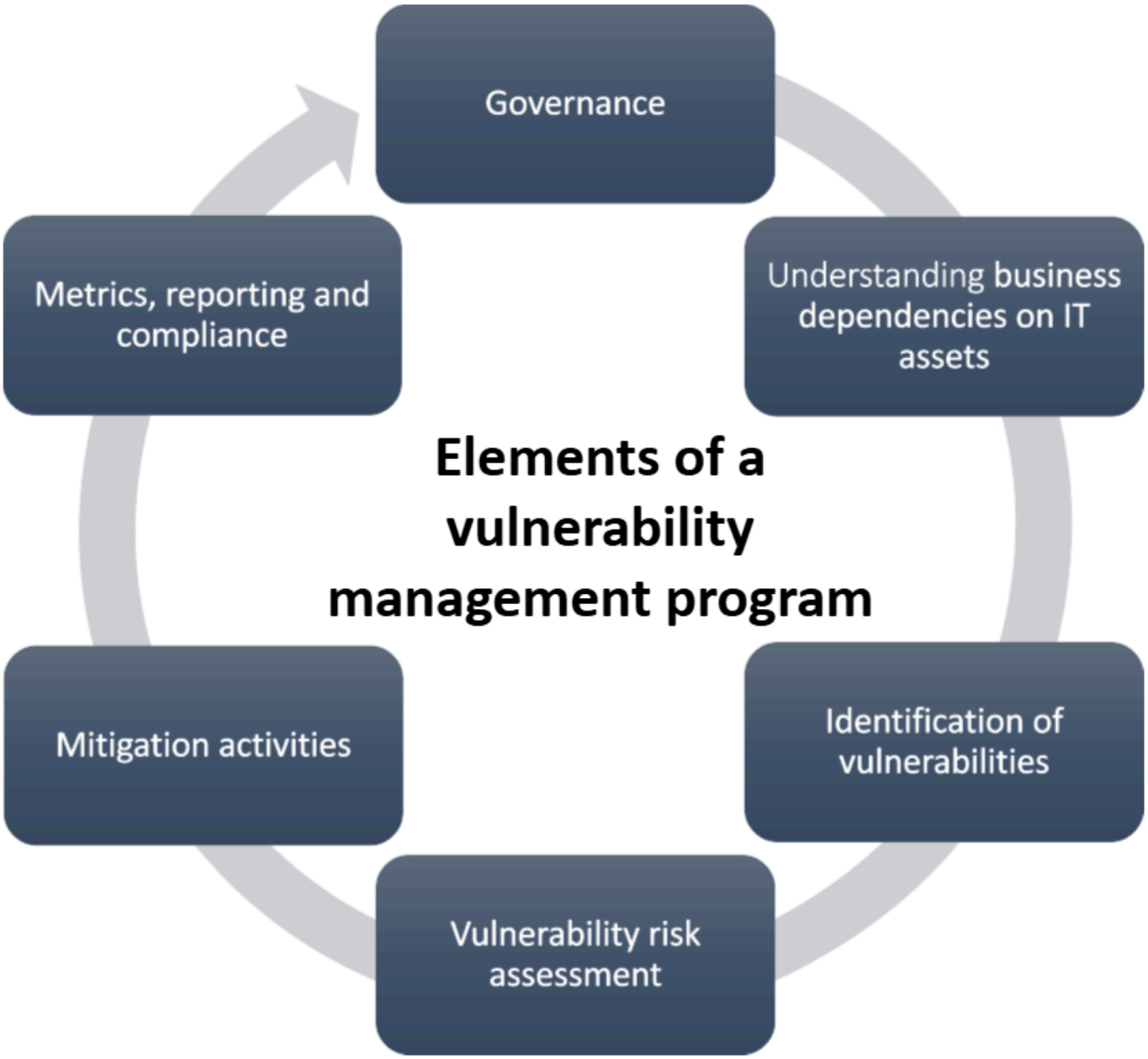
The target audience for this guidance is individuals responsible for the management, analysis and remediation of vulnerabilities that affect the GC's IT systems. This includes cyber security risk managers, VM teams, IT service owners, IT service administrators and IT security operators.

2. Vulnerability management overview

▼ In this section

- 2.8 Technology considerations

Figure 2.1: Elements of a vulnerability management program



The essential elements of an effective VM program include the following:

- **Governance:** Well-defined policy instruments and governance processes serve as the foundation for all VM activities. Efficient communication and coordination among various teams, including IT, security and development, are crucial. Regular meetings and collaboration frameworks help bridge gaps between teams, fostering a unified approach to security issues.
- **Understanding business dependencies on IT assets:** Effective VM requires a comprehensive and current knowledge of the state of IT assets (that is,

inventory and configurations) and the precise way they enable, and may pose risks to, business activities.

- **Identification of vulnerabilities:** Various methods such as network scans, agents, penetration testing, attack surface management, software component analysis, as well as vendor- and intelligence-based information are crucial for identifying vulnerabilities in an organization's systems. A formal coordinated vulnerability disclosure (CVD) pathway further adds to this understanding.
- **Vulnerability risk assessment:** Vulnerabilities must be prioritized based on the level of risk they pose to business activities. A risk-based approach should be used to assess the severity and potential impact of vulnerabilities, focusing efforts on the most critical threats. A risk framework must provide a transparent, rational and repeatable method for prioritizing vulnerabilities and for determining the associated mitigation time frames
- **Mitigation activities:** The risk assessment process should identify work requirements in the form of patching and other mitigations. Clear timelines for vulnerability remediation should be defined based on risk levels, ensuring timely mitigation of high-risk vulnerabilities and maintaining compliance with regulatory requirement. This work should be facilitated through ticketing, change management, configuration management and patch management tools.
- **Metrics, reporting and compliance:** Robust metrics and reporting on VM activities is crucial for continuous improvement and accountability.

2.1 Governance

Effective VM relies on well-defined policy instruments and governance processes. These establish requirements (CISA), expectations, accountabilities, frameworks and work processes, all approved by executive management to ensure institutional support and resources. Governance documentation should include both high-level policies and tactical procedures aligned with industry best practices. In case of disputes or audits, these documents provide a record of the organization's commitment to structured and transparent governance.

Key elements of VM governance include the following:

- **Leadership commitment:** Strong executive leadership (for example, departmental chief information officer, chief information security officer) support is crucial for the success of a VM program. Leaders must champion the program, allocate resources, and ensure that security is a priority across the organization. Leadership commitment sends a clear message to employees that security is important, encouraging them to take ownership and participate in the program.
- **Decision-making bodies and protocols:** Clarify who has the authority and responsibility to make decisions, ensuring alignment with the organization's risk management objectives.
- **Risk assessment and mitigation:** Define processes and frameworks for assessing vulnerabilities and establishing mitigation activities and timelines based on risk levels.
- **Performance metrics:** Use metrics to measure various aspects of the VM program to serve as key performance indicators for continuous improvement.
- **Compliance and documentation:** Ensure adherence to external regulations and internal policies, minimizing legal and operational risks. Documentation helps communicate necessary processes and ensures that all stakeholders understand their roles and responsibilities.
- **Compulsory versus recommended elements:** Clearly distinguish between mandatory and recommended elements in the governance documentation.
- **Service level agreements:** Specify expectations and service level agreements for VM services, especially when relying on third parties.

2.2 Understanding business dependencies on IT assets

Understanding the dependencies of critical business activities on specific IT assets is essential for VM. Organizations need a comprehensive awareness of the business activities that generate value and drive their mission. This includes addressing the unique challenges of managing vulnerabilities in cloud environments. Specialized scanning techniques and security controls tailored to dynamic and shared cloud resources should be implemented. To manage vulnerabilities effectively, organizations should map their IT assets to business activities, identifying the business risk associated with each IT asset. The Canadian Centre for Cyber Security's Suggested Security Controls and Control Enhancements (ITSG-33)

controls CM-8 and PM-5 emphasizes the need for methodical mapping of business activities to the underlying IT assets, ensuring that vulnerability prioritization considers both technical urgency and risk to critical business activities.

Mapping IT assets to business activities requires maintaining accurate visibility of the technology ecosystem through comprehensive asset discovery, dependency mapping, real-time inventory tracking, and integration with the organization's change and configuration management processes and tools. Organizations must maintain up-to-date inventories and configuration management databases for their physical and virtual computing assets, including operational technology and Internet of Things assets.

IT asset management and configuration management together allow organizations to track non-compliant software deployed in IT environments. Given the complexity and multi-faceted nature of IT systems, it is essential to include all computing environments in a VM program, whether mobile, on-premises, cloud or operational technology. This includes, but is not limited to, endpoints, applications, middleware, libraries, operating systems, hypervisors, virtual machines, container engines, containers, firmware, hardware, industrial control systems, supervisory control and data acquisition systems, distributed control systems, programmable logic controllers, remote terminal units, Human-Machine Interfaces, sensors, and actuators.

The Policy on Service and Digital mandates the necessity of asset management. For further information on IT asset management, refer to Cyber Centre's Using Information Technology Asset Management (ITAM) to Enhance Cyber Security (ITSM.10.004) (2023).

2.3 Identification of vulnerabilities

Vulnerability identification is a crucial element of VM and involves searching for vulnerabilities within networks under the department's responsibility. Organizations gain knowledge of vulnerabilities by synthesizing data from multiple sources, such as vulnerability scanning, threat intelligence, vendor notifications, penetration testing, and CVD. Integrating threat intelligence feeds helps to stay updated on newly disclosed vulnerabilities and emerging threats, allowing for more valuable prioritization and addressing of vulnerabilities. Each source offers unique

insights, and collectively they provide a comprehensive view for identifying vulnerabilities.

Organizations must perform regular and automated vulnerability scans to continually identify potential vulnerabilities and other security risks within their networks in accordance with the Directive on Security Management. Automated tools should be configured to run scans at regular intervals and after significant changes to the network infrastructure. Up-to-date VM data is essential for supporting cyber incident management, as it allows for accurate assessment of the potential or actual impacts of an incident.

> **Reducing vulnerabilities by using a software development life cycle**
>
> "A **software development life cycle** (SDLC) is a formal or informal methodology for designing, creating, and maintaining software (including code built into hardware). There are many models for SDLCs, including waterfall, spiral, agile, and – in particular – agile combined with software development and IT operations (DevOps) practices… Regardless of which SDLC model is used, secure software development practices should be integrated throughout it for three reasons: to reduce the number of vulnerabilities in released software, to reduce the potential impact of the exploitation of undetected or unaddressed vulnerabilities, and to address the root causes of vulnerabilities to prevent recurrences. Vulnerabilities include not just bugs caused by coding flaws, but also weaknesses caused by security configuration settings, incorrect trust assumptions, and outdated risk analysis." (Source: NIST SP 800-218, Secure Software Development Framework V1.1: Recommendations for Mitigating the Risk of Software Vulnerabilities)

Commercial VM platforms typically cannot detect flaws in custom business applications or government off-the-shelf applications. The risk-managed life cycle of these applications should include periodic targeted vulnerability assessments, penetration testing, and a CVD process to identify software flaws and configuration deficiencies.

Custom applications often rely on underlying commercial-off-the-shelf or open-source components, application programming interfaces, packages or frameworks. Application owners should be aware of these dependencies, including using a software bill of materials (SBOM) and solution architecture documentation. An SBOM is a comprehensive inventory of all software components, including libraries, tools and processes used in the development of a software product. The SBOM plays a crucial role in vulnerability management by providing transparency into the software supply chain, enabling organizations to quickly identify and assess potential vulnerabilities. When paired with software composition analysis (SCA) tools, SBOMs can significantly enhance security by scanning open-source dependencies for known vulnerabilities, risky licences and malware.

2.3.1 Identification methods

There are multiple methods to identify vulnerabilities within systems, and they include both automated and manual assessment methods. The following are some of the most common approaches used in VM.

2.3.1.1 Internal assessments

Organizations regularly assess for vulnerabilities using specialized tools or processes. Automated scanning tools scan predefined Internet Protocol (IP) ranges or use agents installed on systems to identify known vulnerabilities. Although agents allow for continuous scanning, not all assets (for example, routers, serverless computing, multi-function devices) can support agent installation. Additionally, in some environments, installing agents may not be feasible or permitted. Scans can be performed on production assets or in a lab environment, with results extrapolated to the production environment. It is crucial to ensure that vulnerability signature files are automatically and regularly updated.

Internal assessment methods include the following:

- **vulnerability scanning:** automated tools that scan systems for known vulnerabilities based on predefined IP ranges or installed agents
- **configuration checking:** tools that assess system and network configurations to ensure compliance with security best practices and organizational policies

- **infrastructure as code (IaC) scanning:** analyzing code that defines infrastructure (for example, Terraform, CloudFormation) to identify security issues before deployment
- **patch management tools:** systems that track and manage the application of patches to software and hardware to address vulnerabilities
- **endpoint detection and response (EDR):** solutions that monitor and respond to threats on endpoints, providing continuous assessment and remediation
- **compliance audits:** regular checks to ensure systems and processes comply with internal policies and external regulations
- **log analysis:** reviewing system and application logs to identify unusual activities that may indicate vulnerabilities or breaches
- **network traffic analysis:** monitoring network traffic to detect anomalies that could signify vulnerabilities or malicious activities
- **penetration testing:** simulating real-world attacks to uncover vulnerabilities that automated tools might miss

2.3.1.2 External assessments

External assessments involve identifying vulnerabilities from outside the organization's network perimeter. This approach helps to understand how an attacker might exploit weaknesses from an external perspective. Tools and services for attack surface management fall into this category, helping organizations map and monitor their external-facing assets. External assessment tools simulate an attacker's perspective by assessing the organization's external IP addresses and domains for vulnerabilities. Regular external assessments should be performed, especially after significant changes to the network infrastructure.

External assessment methods include the following:

- **attack surface management:** tools and services that help organizations map and monitor their external-facing assets to identify vulnerabilities
- **external vulnerability scanning:** tools that simulate an attacker's perspective by scanning the organization's external IP addresses and domains for vulnerabilities
- **third-party assessments:** engaging external security firms to perform independent assessments of the organization's external security posture

- **coordinated vulnerability disclosure (CVD):** Implementing a CVD process that ensures stakeholders and partners are made aware of vulnerabilities that could impact them. This approach leverages the expertise of the community to identify and address security issues. CVD programs require clear guidelines and processes for external reporting and timely communication with researchers.

2.3.1.3 Incident response

Incident response plays a role in enhancing VM programs. Post-incident insights gained through review and analysis help understand how a breach occurred and what vulnerabilities were exploited. This analysis provides valuable information about weaknesses in the system, helping to prioritize the remediation of vulnerabilities that have been actively exploited. Defined procedures for dealing with vulnerability disclosures improve the ability to react to and remediate vulnerabilities. The Government of Canada Cyber Security Event Management Plan (GC CSEMP) includes vulnerabilities as part of the definition of a cyber security event, which is any event, act, omission or situation that may be detrimental to government security. Departments are expected to establish a departmental cyber security event management plan that is aligned with the GC CSEMP.

2.3.2 Suggested assessment time frames

To ensure timely and effective identification of vulnerabilities, organizations should adopt a structured schedule for each assessment method. Table 2.1 outlines recommended frequencies for various vulnerability identification techniques, based on GC guidance and industry best practices. These timelines are intended to support proactive risk management and align with evolving threat landscapes and operational needs.

Table 2.1: Suggested assessment time frames

Identification method	Recommended frequency	Rationale / Source
Vulnerability scanning (internal)	At least weekly or after significant changes;	Canadian Centre for Cyber Security: *IT Security Risk Management: A Lifecycle Approach (ITSG-33)* RA-5;

	agents for continual assessment	aligns with proactive detection and patch cycles
Vulnerability scanning (external)	Monthly and after major infrastructure changes	Supported by the Canadian Centre for Cyber Security's Security Considerations for Edge Devices (ITSM.80.101) ITSM.80.101 and industry best practices.
Configuration checking	Monthly and after configuration changes	Aligns with *IT Security Risk Management: A Lifecycle Approach (ITSG-33)* CM-6 and CM-8 controls
IaC scanning	At every code commit or pull request	DevSecOps best practice; ensures vulnerabilities are not introduced before deployment
Patch management tools	Continuous monitoring; review weekly	Ensures timely patch application and compliance with GC patching guidance
EDR monitoring	Continuous	Real-time detection and response; Canadian Centre for Cyber Security Top 10 IT Security Actions
Compliance audits	Semi-annually	Based on GC audit cycles and *IT Security Risk Management: A Lifecycle Approach (ITSG-33)* PM-5
Log analysis	Daily or near real time	Based on *IT Security Risk Management: A Lifecycle Approach (ITSG-33)* AU-6
Network traffic analysis	Continuous	Detects anomalies and potential vulnerabilities in real time
Penetration testing	Annually or after major system changes; more frequently for higher-risk systems	Based on *IT Security Risk Management: A Lifecycle Approach (ITSG-33)* CA-8
Attack surface management	Continuous with monthly reviews	Continuous monitoring of external exposure maintains visibility of

		Internet-facing assets and exposures
Third-party assessments	Annually or as needed based on risk	Ensures independent validation of security posture
Coordinated vulnerability disclosure	Ongoing; review reports within five business days of receipt	GC CSEMP and industry best practices for timely response to external reports

2.4 Vulnerability risk assessment

GC organizations are **required** to "analyze impacts of identified vulnerabilities, and implement corrective actions (for example, apply patches and updates, in accordance with defined timelines and, as required, on an emergency basis)." (Source: Directive on Security Management, Appendix B, subsection B.2.3.7.3)

Effective risk assessment is the cornerstone of any VM program, enabling organizations to prioritize mitigation activities to protect the continued delivery of business activities. The key method for this is a vulnerability risk assessment calculation tool, which should be designed based on the following principles:

- **Clearly defined risk factors:** Risk factors should be clearly defined and distinguishable from one another. Examples include the ease of exploitation, active exploitation in the wild, presence on IT systems supporting critical business operations, and whether vulnerabilities are on Internet-connected IT assets. It should also be indicated whether there have been existing mitigation or compensation controls implemented for each factor.
- **Significant value in risk determination:** Risk factors should add significant value in determining risk.
- **Unambiguous scoring guide:** A scoring guide should explain in unambiguous language how different scores are assessed for each risk factor.
- **Transparent scoring mechanism:** The scoring guide should also explain how the scores of each risk factor are combined to arrive at an overall risk score. This process should be transparent, straightforward and easy to understand.

A vulnerability risk assessment method should produce two critical pieces of analysis:

1. **Overall risk score:** Indicates the level of risk associated with a vulnerable asset.
2. **Mitigation timelines:** Specific timelines for mitigation linked to risk scores, decreasing in duration as the risk increases. These timelines should align with those of similar organizations and be effective against known threat actor exploitation speeds, as reported by vendors and intelligence sources.

There are many ways to conduct a risk assessment. For instance, the Cybersecurity and Infrastructure Agency (CISA) has released the Stakeholder-Specific Vulnerability Categorization (SSVC), which is an example of using decision trees. Weighted average risk calculations are another method for conducting assessments. Appendix C provides a vulnerability risk assessment approach using weighted average risk scoring that considers the following risk factors:

- vendor risk rating
- vulnerability technical rating
- exploit maturity
- asset importance to threat actors
- asset importance to business activities
- asset external exposure

While adherence to mitigation timelines is crucial, there can be exceptions and exemptions to patching. For vulnerabilities without available patches, alternative mitigations (for example, network segmentation) or temporary workarounds (for example, disabling the vulnerable feature) may serve as interim measures. In non-zero-day situations, alternative mitigations can also be acceptable, but they must not leave the organization exposed to additional risk. Any exception to the established patching timeline should be backed by a risk assessment to confirm that the alternative measures appropriately mitigate the risk.

In some cases, risk owners may decide to accept certain risks associated with vulnerabilities. A formal risk acceptance process and risk acceptance registry are pivotal for managing vulnerabilities that cannot be immediately mitigated. This process involves documenting instances where an organization decides to accept the risk associated with a specific vulnerability rather than implementing immediate corrective actions. The risk acceptance registry should detail:

- the rationale for each acceptance decision, including an analysis of the impact versus the cost of mitigation, and whether there are controls in place that partially mitigate the risk
- the conditions under which the risk acceptance will be reviewed and potentially revisited, such as changes in the threat landscape or the availability of more effective mitigation strategies

This registry serves as a critical tool for tracking accepted risks over time, ensuring they are acknowledged, justified, and continuously assessed against organizational risk tolerance and security posture. Any risk acceptance must include an expiry date that is less than 12 months from when it is issued. This ensures a regular review and re-acceptance.

2.5 Mitigation activities

Vulnerability mitigation can take several forms, depending on the nature and severity of the vulnerability, the complexity and criticality of the vulnerable system, and even the availability of a patch. Although it may not always be possible, the preferred method is to apply the vendor-supplied patch or adjust the configuration setting to fully remediate the issue.

2.5.1 Patching

Patches are updates made to firmware and software to correct functional and security deficiencies. Applying patches to operating systems, applications and devices is critical for system security. Leveraging automation to assist with patching can significantly enhance efficiency and accuracy, reduce the risk of human error, and ensure timely application of critical updates. Emergency patches, often in response to zero-day vulnerabilities, should follow a well-defined emergency change schedule.

Patches and updates must be tracked through the departmental change management system (Directive on Security Management, subsection B2.3.3). Patch application plans should include contingency and rollback plans. More detailed information on patch management can be found in Security Vulnerabilities and Patches Explained: IT Security Bulletin for the Government of Canada (ITSB-96).

2.5.2 Configuration changes

A common and crucial vulnerability mitigation involves configuration changes and should be included in configuration and change management processes. Configuration changes involve adjusting system settings to enhance security and mitigate vulnerabilities. This is typically referred to as a hardened configuration, according to CCCS's Top 10 IT Security Actions to Protect Government of Canada Internet-Connected Networks and Information (ITSB-89 v3). Features typically enabled by default should be disabled to minimize exposure and reduce the need for emergency patches if they are not required. Hardening can also include disabling or removing non-required accounts and services, changing the manufacturer's default passwords, as well as modifying access controls. Configuration changes should be documented and managed through the configuration management process to ensure they are implemented correctly and consistently. Regular reviews and audits of configurations can help maintain security and compliance.

2.5.3 Compensating controls

Full mitigation of the vulnerability is the preferred option when addressing vulnerabilities. When patching or configuration changes are not immediately feasible, compensating controls provide alternative means to maintain security and assist with risk reduction. The goal of compensating controls is to enable business continuity while still addressing security concerns. Often, multiple compensating controls are used together to bring the risk to an acceptable level until a permanent fix is possible.

Examples of compensating controls (non-exhaustive listing):

- **network security zoning:** segmenting the network to isolate vulnerable systems and limit potential attack vectors
- **web application firewalls:** protecting web applications by filtering and monitoring HTTP traffic between a web application and the Internet
- **intrusion detection and prevention systems:** monitoring network or system activities for malicious activities or policy violations and taking preventive actions

- **disabling or removing vulnerable services or software:** temporarily disabling or uninstalling services or software that are known to be vulnerable
- **access controls:** implementing strict access controls to limit who can access sensitive systems and data
- **application allow-listing:** allowing only approved applications to run on the network, reducing the risk of malicious software execution
- **security awareness training:** educating employees about security best practices and how to recognize potential threats
- **increased monitoring:** enhancing monitoring efforts to detect and respond to any unusual activities or exploitation attempts promptly

Compensating controls should be reviewed and re-approved at least yearly to ensure they remain effective and aligned with the evolving threat landscape

2.5.4 Cultural support for mitigation

Effective mitigation activities rely on a supportive organizational culture. Strong leadership commitment ensures that security is prioritized and resources are allocated appropriately. Leaders should actively participate in security discussions and hold teams accountable for security performance.

Collaboration between IT, security, development and other departments is crucial. Regular meetings to discuss security requirements and coordinate remediation efforts foster a unified approach. A security-conscious culture encourages employees to follow best practices and take ownership of their security responsibilities. Providing security awareness training and promoting incident reporting are key components.

Open communication and transparency about security issues build trust and cooperation among teams. Regular updates on mitigation activities and clear documentation of processes are essential. Continuous improvement ensures that mitigation strategies evolve to address new threats. Regularly reviewing and updating processes, incorporating feedback, and investing in training are vital practices.

2.6 Metrics, reporting and compliance

A VM program should include continuous monitoring and improvement to maintain security posture against evolving threats. This includes using tools and techniques to monitor systems, perform regular assessments, and apply timely updates and patches. Metrics should evaluate the coverage, effectiveness, efficiency and compliance of the VM activities. Regular reporting on metrics will enable the organization to establish a baseline of performance. Metrics should be informed by IT service management tools so that vulnerabilities are ticketed and tracked for mitigation. Furthermore, metrics should link to a VM maturity model, which allows for a rational and transparent approach to measuring the maturity of the VM program. Maturity models assist program managers in identifying which areas are performing as desired and which areas require additional reform and investment.

Reports and dashboards enable continuous monitoring and compliance evaluation. These tools provide a snapshot of the organization's security posture at any given moment, allowing for prompt response to emerging vulnerabilities and demonstrating adherence to regulatory requirements.

Table 2.2: Sample vulnerability management metrics

Category	Metric
Coverage	• **Inventory coverage:** the percentage of hardware and software within the organization that is regularly inventoried and evaluated for vulnerabilities
Efficiency and effectiveness	• **Patch time metrics:** the minimum, average and maximum time to patch a given percentage of hosts • **Mean time to remediate:** the average time between the detection of a vulnerability and its remediation, categorized by risk level • **Patch method distribution:** the percentage of hosts patched automatically, partially (in the case of patches bundled in a package), and manually
Compliance	• **Mitigation compliance:** vulnerabilities remediated outside of the required mitigation period • **Patching exceptions and exemptions:** the number of exceptions and exemptions to patching

Appendix D provides additional suggested metrics to support departmental VM programs.

2.7 Enabling processes

VM does not operate in isolation; it relies on a set of foundational processes that provide essential support and infrastructure. These enabling processes are critical for ensuring that vulnerabilities are identified, assessed and mitigated in a systematic and efficient manner. They create a robust framework that enhances the overall security posture of the organization by integrating various aspects of IT and security management.

The GC Enterprise Cyber Security Strategy GC Enterprise Cyber Security Strategy Annex A: Target Security Operating Model, outlines some key enabling processes that underpin a successful VM program, highlighting their roles and how they contribute to the identification, prioritization and remediation of vulnerabilities. By understanding and implementing these processes, organizations can build a comprehensive and resilient approach to managing vulnerabilities and protecting their assets.

- **Change management:** Change management ensures that all changes to IT systems are managed systematically by implementing policies and tools to track and document changes.
- **Patch management:** Patch management addresses software and hardware vulnerabilities through timely patches, regularly reviewing and applying them using automated tools.
- **Configuration management:** Configuration management maintains secure IT system configurations by using tools to enforce secure settings and regularly updating configurations.
- **Access management:** Access management controls access to IT systems to prevent unauthorized access by implementing role-based access controls and using multi-factor authentication.
- **Threat intelligence:** Threat intelligence involves staying informed about emerging threats and vulnerabilities by integrating threat intelligence feeds and regularly reviewing reports.

- **Vendor management:** Vendor management ensures that third-party vendors comply with security standards by conducting regular security assessments and including security requirements in contracts.
- **Audit and compliance:** Audit and compliance ensure that the organization complies with security regulations and standards by conducting regular audits and addressing any compliance gaps.
- **Training and awareness:** Training and awareness educate employees on security best practices and VM by developing a training program, conducting regular sessions, and providing role-specific training.

Refer to GC Enterprise Cyber Security Strategy Annex A: Target Security Operating Model for more information.

2.8 Technology considerations

For detailed information regarding the technology options and capabilities of vulnerability identification platforms, it is recommended to review NIST SP 1800-31: Improving Enterprise Patching for General IT Systems: Utilizing Existing Tools and Performing Processes in Better Ways, 1800-31B (April 2022).

3. Conclusion

A successful VM program is essential for safeguarding an organization's IT assets and ensuring the continuity of business activities. This guidance document aims to help organizations achieve a robust and comprehensive vulnerability management program. It has outlined the key elements of VM governance, including decision-making protocols, risk assessment and mitigation strategies, performance metrics, compliance requirements, and service level agreements. Additionally, the importance of leadership commitment, cross-functional collaboration, and a security-conscious culture has been emphasized as foundational to the effectiveness of the VM program.

Appendix A: Definitions

attack surface management

Attack surface management is the process of continuously identifying, cataloguing and securing all digital elements, services, and endpoints within an organization that are exposed to potential attackers. The goal is to systematically catalogue and reduce the vulnerability of attackable points within an organization's digital infrastructure to mitigate the risk of cyber threats.

coordinated vulnerability disclosure

Coordinated vulnerability disclosure is the process of systematically managing the reporting, analysis and mitigation of cybersecurity vulnerabilities. The goal is to ensure that vulnerabilities identified by researchers, users or other stakeholders are reported directly to the organization, allowing for the development and deployment of necessary patches or mitigations before public disclosure. This process helps balance the need for timely vulnerability disclosure with the imperative to protect users by addressing issues before they are widely known.

critical systems and services

Critical systems and services are defined by TBS as those "whose compromise in terms of availability or integrity would result in a high degree of injury to the health, safety, security, or economic well-being of Canadians or to the effective functioning of the Government of Canada" (Source: Policy on Government Security)

patch management

The process of managing a network of computers by regularly performing system updates and patches to ensure that systems are protected against vulnerabilities and security threats. [3]

penetration testing

Penetration testing, or pen testing, is a simulated cyber-attack against a system to check for exploitable vulnerabilities. It is a proactive and authorized attempt to evaluate the security of a system by safely trying to exploit system vulnerabilities.

software bill of materials

A software bill of materials (SBOM) is a formal record that details the components and supply chain relationships of the various elements used to build a software package. It functions as a nested inventory, listing all the ingredients that comprise the completed software solution. An SBOM provides transparency into the software's composition, helping organizations identify and manage vulnerabilities, ensure compliance, and enhance security throughout the software development life cycle.

network scans
Network scans involve the use of software tools to identify devices, servers and other entities on a network, along with their services, configurations and vulnerabilities. This is a critical step in assessing the security posture of networks.

threat intelligence
Information about threats and threat actors that helps mitigate harmful events in cyberspace. It includes details about mechanisms, indicators, implications and actionable advice about existing or emerging threats.

vulnerability
A vulnerability is a weakness or flaw in an information system, system security procedures, internal controls or implementation that could be exploited by a threat actor to gain unauthorized access to information or disrupt critical services. Vulnerabilities can arise from various sources, including software bugs, misconfigurations or inadequate security practices.

Appendix B: Vulnerability Management Checklist

▼ In this section

The following is a checklist that summarizes the core components discussed in this document. Following this structure will help to establish a clear and effective vulnerability management program that aligns with organizational goals and protects critical assets.

B-1: Vulnerability Management Checklist

1. Governance

Objective: Establish a formal structure to manage vulnerabilities and ensure accountability.

Steps

- Define the scope and objectives of the program.
- Assign roles and responsibilities (for example, security team, asset owners, executive sponsors).
- Develop policies and procedures to guide vulnerability management efforts.
- Ensure compliance with GC policies, standards and regulatory requirements.

2. Understanding business dependencies on IT assets

Objective: Identify critical assets and their dependencies

Steps

- Inventory all IT assets, including hardware, software and network components.
- Map business processes to the IT assets they depend on.
- Prioritize assets based on their criticality to business operations and data sensitivity.

3. Identification of vulnerabilities

Objective: Detect vulnerabilities in IT assets to manage risks proactively

Steps

- Conduct regular internal and external assessments using automated tools.
- Augment automated assessments with manual testing and validation.
- Leverage incident response outcomes to identify flaws.
- Monitor threat intelligence feeds for newly disclosed vulnerabilities.

4. Vulnerability risk assessment

Objective: Evaluate and prioritize vulnerabilities based on their potential impact and likelihood.

Steps

- Assess the severity of identified vulnerabilities.
- Consider the asset's criticality and exposure to threats.
- Create a risk matrix to categorize vulnerabilities by priority (for example, high, medium, low).

5. Mitigation activities

Objective: Remediate vulnerabilities or apply compensating controls to reduce risk until full remediation can be achieved.

Steps

- Patch or update affected systems and applications.
- Apply configuration changes to mitigate risks temporarily.
- Implement compensating controls to reduce risk to an acceptable level when complete remediation is not possible or will take too long to complete.

6. Metrics, reporting and compliance

Objective: Measure program effectiveness and demonstrate compliance

Steps

- Develop key performance indicators (for example, time to remediate, number of high-risk vulnerabilities addressed).
- Generate regular reports for stakeholders, highlighting trends and achievements.
- Ensure alignment with regulatory and audit requirements through documented evidence.

B-2: Detailed Program Execution Checklists

This section provides additional checklists to support effective program execution and continuous improvement. They address governance, operational execution, and integration with broader IT security functions. Departments may use them to conduct self-assessments of their departmental VM program's maturity and alignment with the Guideline on Vulnerability Management.

Table B.1: Policy and Governance Foundations Checklist

Checklist Item	Status (Yes / No / NA)	Notes
Department has a formal Vulnerability Management Policy		
The policy is approved by senior leadership		
Governance structure is documented (roles, escalation paths, decision authorities)		
VM policy includes alignment to GC policy instruments (e.g., ITSG-33, CSEMP, GC Cyber Policy)		
Roles and responsibilities are defined across IT, Security, and Business stakeholders		
VM is included in departmental cybersecurity governance bodies or steering committees		

Table B.2: Asset Inventory and Criticality Mapping Checklist

Checklist Item	Status (Yes / No / NA)	Notes
A current, authoritative inventory of IT assets exists (hardware, software, systems)		
Inventory includes system owner, environment, business function, and sensitivity		
Assets are classified by criticality (e.g., mission-critical, internal, public-facing)		
Externally exposed systems and services are clearly identified and tracked		
Asset data is accessible by the VM program for scoping and prioritization		
CCCS host-based sensors are deployed on all client endpoints and cloud-based sensors are implemented per GC Cloud Guardrails		

Table B.3: Vulnerability Detection and Monitoring Checklist

Checklist Item	Status (Yes / No / NA)	Notes
Internal scans are conducted at least weekly (or per documented schedule)		
External scans are conducted at least monthly		
Scanning covers the full authorized asset inventory		
Scanner outputs are integrated with asset inventory, ticketing, and/or SIEM platforms		
Zero-day and emerging threats are tracked via vendor feeds, CISA KEV (Known Exploited Vulnerabilities), EPSS (Exploit Prediction Scoring System), and CCCS alerts		
Coordinated Vulnerability Disclosure (CVD) process is defined and monitored		

Table B.4: Risk Assessment and Prioritization Checklist

Checklist Item	Status (Yes / No / NA)	Notes
A documented risk assessment methodology is used (e.g., Weighted Average or Decision Tree)		
Factors such as CVSS, exploitability, asset importance, and exposure are considered		
CISA KEV, EPSS, or threat intelligence enrich risk assessments		
High-risk vulnerabilities are identified, tracked, and escalated per policy		
Risk tolerance thresholds and decision-making logic are documented and reviewed regularly		

Table B.5: Risk Remediation and Risk Treatment Checklist

Checklist Item	Status (Yes / No / NA)	Notes
Remediation timelines are documented (e.g., 2/14/60/90 days by risk level)		
SLA targets are tracked and reported		
Risk acceptance is formally documented with executive sign-off		
Compensating controls are evaluated and documented when remediation is not feasible		
Change management processes support emergency patching and remediation		

Table B.6: Verification and Validation Checklist

Checklist Item	Status (Yes / No / NA)	Notes
Rescanning is conducted after remediation to verify closure		
Validation includes functional testing of patches or mitigations		
Evidence of remediation and validation is retained (e.g., logs, tickets, screenshots)		
Closure of findings is documented in ticketing or reporting systems		

Table B.7: Metrics and Reporting Checklist

Checklist Item	Status (Yes / No / NA)	Notes
Metrics are defined (e.g., MTTR, coverage %, open findings, SLA compliance)		

Dashboards or regular reports are shared with key stakeholders (e.g., IT Ops, Execs)		
Metrics are presented to cybersecurity governance committees		
Trends, gaps, and systemic issues are identified using metrics and inform program updates		

Table B.8: Integration with Incident Response and IT Security Processes Checklist

Checklist Item	Status (Yes / No / NA)	Notes
The VM process is integrated with the Incident Response Plan (IRP)		
CCCS threat intelligence (e.g., Cyber Flashes) and TBS directives (e.g., Requests for Action) are incorporated into the VM process		
Department subscribed to the CCCS National Cyber Threat Notification System (NCTNS)		
High-risk vulnerabilities can trigger IR actions (e.g., Indicator of Compromise searches, threat hunts)		
Roles and communication flows during exploitation events are documented		
Vulnerability findings inform threat modeling, control assessments, and IT project reviews		
Departmental Cyber Security Event Management Plan (CSEMP) includes reference to VM inputs and actions		

Table B.9: Continuous Improvement and Audit Readiness Checklist

Checklist Item	Status (Yes / No / NA)	Notes

VM plan and VM policy are reviewed annually or after significant events		
Lessons learned from incidents or audits are documented and drive improvements		
Internal (e.g., evaluation against a VM maturity framework) or third-party reviews of the VM program are conducted periodically		
VM stakeholders receive role-specific training (e.g., analysts, system owners)		
Program updates are informed by emerging technologies, GC guidance, and real-world incidents		

Appendix C: Vulnerability Risk Assessment Tool

▼ In this section

The purpose of this Vulnerability Risk Assessment Tool is to:

- have a rational, justifiable and easy-to-use approach to prioritize vulnerabilities according to the risk they pose to the organizational cybersecurity posture and the delivery of critical services
- identify vulnerability remediation expectations that are in line with similar organizations and account for threat actor exploit development speed
- have a consistent approach across the GC

The Vulnerability Risk Assessment Tool adheres to industry best practices. As a general rule, GC organizations should prioritize **critical systems and services** in any mitigation scenario, and the higher potential risk to these systems should be

accounted for in the vulnerability risk assessment tool. A comprehensive understanding and cataloguing of critical systems are also crucial for business continuity and IT disaster recovery.

C-1: Process

The following steps should be applied when using the tool:

1. Score each risk factor.
2. Add up the scores (total out of 50 points).
3. Identify where the score is on the risk matrix for testing and remediation time frames.

Considerations when using the tool:

- Subject matter expertise is required to score factors.
- If a factor cannot be scored, then calculate the sum of other factor scores as a percentage of available points and map as follows:
    - red = 80% to 100%
    - orange = 60% to 79%
    - yellow = 40% to 59%
    - green = 1% to 39.

C-2: Risk factors

Table C.1 outlines the risk factors:

Table C.1: Risk Factor Chart (total of 50 points)

Factor	Description	Maximum score
Vendor risk rating	Provided by vendor	5
Technical exploitability and technical impact (Common Vulnerability Scoring System base score)	Provided in National Vulnerability Database	10

Exploit maturity	Degree to which exploit is available and used in the wild	10
Asset importance to threat activities	Importance of vulnerable asset to threat actor because of what the threat activity asset will enable	5
Asset importance to business activities	Importance of vulnerable asset to business based on business, application or system risk register	10
Asset external exposure	Extent to which vulnerable asset is Internet accessible	10

C-3: Risk factor scoring guides

The following section outlines scoring guides for several risk factors. As noted in Table C.1, the vendor risk rating and Common Vulnerability Scoring System base score are made available by the vendor and National Vulnerability Database.

Table C.2: Scoring guide for exploit maturity

Points	Description
8 to 10	Functional autonomous code exists or no exploit is required (manual trigger), and details are widely available. Exploit code works in every situation or is actively being delivered via an autonomous agent (such as a worm or virus). Network-connected systems are likely to encounter scanning or exploitation attempts. Exploit development has reached the level of reliable, widely available and easy-to-use automated tools.
5 to 7	Functional exploit code is available. The code works in most situations where the vulnerability exists.
3 to 4	Proof-of-concept exploit code is available, or an attack demonstration is not practical for most systems. The code or technique is not functional in all situations and may require substantial modification by a skilled attacker.

1 to 2	No exploit code is available, or an exploit is theoretical.

Table C.3: Scoring guide for asset importance to threat activities

Points	Description

4 to 5	• Asset contains sensitive information, which can lead to data exfiltration (for example, database injection flaws, unencrypted data storage, exposed application programming interface (APIs).
	• Asset hosts enterprise services where lack of availability has ancillary impacts (for example, time service, Domain Name Service (DNS), federated services, certificate authority).
	• Asset provides initial access to an attacker, leading to potential intrusion (for example, exposed remote services, insecure public-facing web applications, open ports, VPN vulnerabilities).
	• Asset contains an identity store that allows an attacker to escalate privileges via:
	  ◦ control of accounts, which allows access to the network infrastructure or endpoints with administrator privileges
	  ◦ control of accounts used to access sensitive data
	  ◦ control of service accounts
	  ◦ control of accounts used for API calls
	  ◦ control of accounts that allow access to the network environment

3	• Asset allows an attacker to gain access, without authentication, to local resources with user privileges.
1 to 2	• Asset allows attackers to move from one asset to another, spreading their influence (for example, network protocol flaws, insecure internal web applications, insecure service accounts). • Asset that can be exploited by attackers to maintain access over time (for example, insecure backup routines, vulnerable firmware, auto-run vulnerabilities).

Table C.4: Scoring guide for asset importance to business activities

Points	Description

8 to 10	Asset is fundamental to core business operations. Compromise of this asset directly impacts business continuity and may have severe financial, operational or reputational repercussions. Higher numbers of impacted assets typically associated with higher score. Can include assets supporting Protected B (PB) systems. Includes assets that support Secret and Protected C information.
5 to 7	Asset aids critical business operations but is not the primary driver. Although its compromise will not halt business operations, it may cause significant disruptions or inefficiencies.
3 to 4	Asset supports secondary business functions that are not tightly intertwined with core operations. Its compromise, although detrimental, can be managed without major disruptions to core functionalities.
1 to 2	Asset aids in non-essential tasks or operations with minimal impact if compromised. Its compromise has minimal immediate impact on primary business functions. It might serve peripheral, non-essential roles and can be readily replaced or bypassed.

Table C.5: Scoring guide for asset external exposure

Points	Description
8 to 10	Vulnerable asset is externally routable. A threat actor could exploit the asset from the Internet.
5 to 7	Vulnerable asset is remotely exploitable. A threat requires access to same Wide Area Network (WAN) as the vulnerable asset to exploit it.
3 to 4	Vulnerable asset is intra-network exploitable. A threat actor requires access to same network segment as the vulnerable asset to exploit it.
2	Vulnerable asset is locally exploitable. A threat actor requires direct access to the vulnerable asset.
1	Vulnerable asset is physically exploitable. A threat actor requires physical access to the vulnerable asset.

C.4: Risk matrix

Table C.6 provides recommended time frames for testing and remediation of identified vulnerabilities.

Table C.6: Risk matrix

Points	Percentage	Overall risk level	Remediation time frame	Process
40 to 50	80% to 100%	Critical	48 hours	GC CSEMP referral; departmental vulnerability remediation
30 to 39	60% to 79%	High	14 days	Departmental vulnerability remediation
20 to 29	40% to 59%	Medium	30 days	Departmental vulnerability remediation
1 to 19	1% to 39%	Low	90 days	Departmental vulnerability remediation

Appendix D: Additional Metrics

Effective vulnerability management requires meaningful, layered metrics that support decision-making across technical, programmatic, and executive levels. Rather than focusing on raw numbers alone, departments should use metrics that contextualize exposure, prioritize risk, and support trending over time.

The following table outlines suggested metrics, how to calculate them, and how to use them to drive insight and accountability. Departments should tailor and extend these to suit their operational realities, reporting structures, and business contexts.

Table D.1: Metrics

Category	Sub-Category	Description
Core Vulnerability Management Metrics	*Not applicable*	• # Servers Scanned: Total servers scanned by VM tool (credentialed and non-credentialed).

		• Coverage: % of IT assets included in regular vulnerability scanning/patch management. (the goal should be 100% of in-scope assets covered). • # Vulnerable Legacy Servers: Servers with end-of-life operating systems. • # Vulnerable Non-Legacy Servers: Servers with vendor-supported OS. • # Servers Scannedwithout Credentials: Indicates incomplete vulnerability data. • # Servers Scanned with Credentials: Core focus for remediation. • # Servers without a Remediation Plan: Excluded due to dependencies or decommissioning. • # Servers with a Remediation Plan: Active remediation scope. • # Total Discovered Vulnerabilities across all scanned servers. • # Total Vulnerabilities on Legacy Servers • # Total Vulnerabilities on Non-Legacy Servers • # Total credentialed scan vulnerabilities • # Total vulnerabilities with a remediation Plan
Layering Metrics for Risk Insight	*Not applicable*	Raw metrics, like the number of open vulnerabilities are insufficient on their own. To make metrics actionable and risk-aligned, departments should consider: • Normalizing by asset volume (e.g., vulns per 100 assets) • Filtering by asset criticality or business function

		• Weighting based on exploitability (e.g., KEV-listed, active exploits) • Removing accepted exceptions from actionable totals • Trending over time to identify improvement or regression Example composite metric: • Critical/KEV Vulnerability Density on High-Impact Assets (per 100 assets), excluding deferred items This layered approach shows organizational risk, not just technical backlog.
Operational-Level Metrics	**Open Vulnerabilities by Age (breakdown by severity)**	**Why it matters:** Tracking open vulnerabilities by age highlights where remediation delays occur, identifies systemic issues in patch management, and helps prioritize older, high-risk vulnerabilities that may expose GC systems to exploitation. It also supports compliance reporting and SLA adherence. **How to calculate:** Difference between first detected and last detected.
	Open Vulnerabilities by Legacy Servers versus Non-Legacy Servers	**Why it matters:** Legacy systems often lack vendor support, making them high-risk and harder to patch. Identifies technical debt and informs migration/decommission plans. **How to calculate:** Filter scan results for servers with end-of-life OS.
	Internet-Facing vs Internal-Facing Assets	**Why it matters:** Internet-facing assets are more exposed and require faster remediation. Supports risk-based

		prioritization and compliance reporting. **How to calculate:** • Internet-Facing = Count of assets with public IP or DNS accessible externally • Internal Facing = Count of assets restricted to internal network
	Total Vulnerabilities by OS, App, Config Layer	**Why it matters:** Helps identify which layer (Operating System, Application, Configuration) contributes most to risk. Enables targeted remediation strategies (e.g., OS patching vs application updates vs hardening). **How to calculate:** From vulnerability scan data, classify each finding by layer: • OS Layer: CVEs tied to operating system (e.g., Windows, Linux). • Application Layer: CVEs tied to installed applications (e.g., browsers, productivity tools). • Configuration Layer: Misconfigurations or missing registry settings.
	Open Vulnerabilities by Severity (Normalized)	**Why it matters:** Known Exploited Vulnerabilities pose the highest risk; attackers actively target them. Prioritization of these vulnerabilities reduces likelihood of compromise. **How to calculate:** Cross-reference scan findings with CISA KEV (or GC Vulnerability List when available)
	Open Vulnerabilities by KEV (Normalized)	**Why it matters:** Shows open vulnerabilities per 100 scanned assets to avoid penalizing larger teams or departments.

		How to calculate: Open Critical CVEs ÷ Total Scanned Assets × 100
	Time to Remediate (TTR)	**Why it matters:** Measures the average or median time taken to remediate vulnerabilities. **How to calculate:** Date Remediated – Date Detected (average/median)
	Exploit Availability Coverage	**Why it matters:** Highlights the portion of known weaponized vulnerabilities that remain unpatched. **How to calculate:** Unpatched KEV CVEs ÷ Total KEV CVEs × 100
	Repeat Findings / Recurrence Rate	**Why it matters:** Flags remediation or configuration weaknesses if vulnerabilities reappear. **How to calculate:** Recurring CVEs ÷ Total fixed CVEs over same period
Program-Level Metrics	**Outstanding Vulnerabilities by Team**	**Why it matters:** Identifies which teams have the largest backlog of vulnerabilities. Helps allocate resources and remediation efforts where they are most needed. **How to calculate:** Outstanding Vulns by Team = SUM (Open Vulnerabilities WHERE Assigned Team = X)
	SLA Compliance Rate	**Why it matters:** Measures how often vulnerabilities are closed within agreed timelines. **How to calculate:** Vulns fixed within SLA ÷ Total fixed vulns × 100
	Backlog Vulnerabilities – Outstanding	**Why it matters:** Indicates SLA breaches and systemic delays.

	Vulnerabilities Older Than 90 Days	**How to calculate:** Count of vulnerabilities where Age > 90 days
	Exception and Exclusion Tracking	**Why it matters:** Tracks how much unresolved risk has been deferred or accepted. **How to calculate:** • Exception Rate = Active Exceptions ÷ Total Open Vulns × 10 • Average Exception Age, Exception Volume by Reason
	Credentialed Scan Coverage (Asset Discovery Effectiveness)	**Why it matters:** Ensures that known assets are being regularly scanned with credentials. **How to calculate:** Regularly Scanned Assets with credentials ÷ Total Known Assets × 100
	Patch Implementation Success Rate	**Why it matters:** Assesses the reliability of the patch process. **How to calculate:** Successful Patches ÷ Patch Attempts × 100
	Reopen Rate	**Why it matters:** Tracks how often vulnerabilities reappear after being marked as remediated. A high reopen rate suggests issues with patch validation, rollback, or configuration management. **How to calculate:** (Number of vulnerabilities that reappear on the same asset within N days ÷ Total number of vulnerabilities marked as resolved in the same period) × 100 *(Common N values: 30, 60, or 90 days)*

Executive-Level Metrics	**Risk Exposure Trend (Composite Score)**	**Why it matters:** Provides a strategic-level view of whether enterprise risk is increasing or decreasing. **How to calculate:** Weighted Scorecard (e.g., KEVs × 3 + Avg TTR × 2 + Open Criticals × 1)
	Business Services at Elevated Risk	**Why it matters:** Links vulnerabilities to business outcomes and service disruption potential. **How to calculate:** Count of services with unresolved critical vulns
	Time to Risk Acceptance or Mitigation Decision	**Why it matters:** Assesses governance responsiveness and risk tolerance. **How to calculate:** Date of decision – Date of detection (average)
	New vs Remediated Vulnerabilities	**Why it matters:** Shows whether remediation is keeping pace with new vulnerabilities. **How to calculate:** For each month: • New Vulns = Count of vulnerabilities first detected in the month • Remediated = Count of vulnerabilities closed in the month • Trend = (Previous Month Total + New) - Remediated

All metrics should be trended monthly or quarterly to show progress, stagnation, or regression. Dashboards, graphs, and heatmaps help make trends more visible to decision-makers.

Where possible, metrics should be broken down by:

- Business Unit
- Business Service / Application

- Technical Owner or Team
- Geography or Classification (e.g., Protected, Classified systems)

This helps identify where risk is concentrated, who is improving, and where accountability should be focused.

Footnotes

1 Treasury Board of Canada Secretariat, Directive on Security Management

2 Canadian Centre for Cyber Security, Top 10 IT Security Actions to Protect Government of Canada Internet-Connected Networks and Information (ITSB-89 v3).

3 Canadian Centre for Cyber Security, Security Vulnerabilities and Patches Explained: IT Security Bulletin for the Government of Canada (ITSB-96)

Date modified: 2026-04-27

Jobs

Immigration and citizenship

Travel and tourism

Business

Benefits

Health

Taxes

Environment and natural resources

National security and defence

Culture, history and sport

Policing, justice and emergencies

Transport and infrastructure

Canada and the world

Money and finances

Science and innovation

Indigenous Peoples

Veterans and military

Youth

Manage life events

Social media