COMPLIANCE COMPONENT

DEFINITION

Name	Enterprise Patch Management
Description	Enterprise patch management is the process of identifying, prioritizing, acquiring, installing, and verifying the installation of patches, updates, and upgrades throughout an organization. Patches, updates and upgrades are pieces of code developed to address security flaws within an application or operating system.
Rationale	Enterprise patch management reduces or eliminates the potential for exploitation.
Benefits	Potentially eliminates the exploitation of vulnerabilities.

ASSOCIATED ARCHITECTURE LEVELS

Specify the Domain Name	Security
Specify the Discipline Name	Management Controls
Specify the Technology Area Name	System Life Cycle Security
Specify the Product Component Name	

COMPLIANCE COMPONENT TYPE

Document the Compliance Component Type	Guideline
Component Sub-type	

COMPLIANCE DETAIL

State the Guideline, Standard or Legislation	*This* document *outlines possible risk response approaches for software vulnerabilities, provides an overview of the software vulnerability management life cycle, and takes a closer look at parts of that life cycle with respect to patching.* **1. Risk Responses -** Patching is one of several ways to respond to risks from software vulnerabilities. **a. Accept** the risk from vulnerable software as is. **b. Mitigate**: Reduce the risk by eliminating the vulnerabilities and/or deploying additional security controls to reduce vulnerability exploitation. **c. Transfer**: Reduce the risk by sharing some of the consequences with another party. **d. Avoid**: Ensure that the risk does not occur by eliminating vulnerable assets.

2. **Software Vulnerability Management Life Cycle** - The following describes a basic software vulnerability management life cycle. This life cycle applies to all risk response approaches.
   - **a.** Know when new software vulnerabilities affect your organization’s assets, including applications, operating systems, and firmware. Know what assets your organization uses and which software and software versions those assets run, down to the level of packages and libraries, as well as keeping track of new vulnerabilities in that software.
   - **b.** Plan the risk response. Assess the risk the vulnerability poses to your organization, choose which form of risk response (or combination of forms) to use, and decide how to implement the risk response.
   - **c.** Execute the risk response. Common phases include the following:
     - i. Prepare the risk response.
     - ii. Implement the risk response.
     - iii. Verify the risk response. Ensure that the implementation has been completed successfully.
     - iv. Continuously monitor the risk response. Ensure that the risk response continues to be in place.
3. **Risk Response Execution –** Below are the common phases of executing a risk response, specifically within the context of patching.
   - **a. Prepare to Deploy the Patch**:
     - **i.** Prioritize the patch. Some patches will have a higher priority than others based on their risk of vulnerability.
     - **ii.** Schedule patch deployment.
     - **iii.** Acquire and validate the patch. After acquiring the patch its authenticity and integrity should be confirmed, preferably by automated means, before the patch is tested or installed.
     - **iv.** Test the patch. Reduce operational risk by identifying problems with a patch before placing it into production.
   - **b. Deploy the Patch** - Common steps for deploying a patch include the following:
     - **i.** Distribute the patch. Distribution may be automatic, manual or scheduled.
     - **ii.** Validate the patch. A patch’s authenticity and integrity should be confirmed before installation, preferably through automated means.
     - **iii.** Install the patch. Installation may be automatic, manual or scheduled.
     - **iv.** Change software configuration as needed in order to properly implement the patch.
     - **v.** Resolve any issues related to the implantation of the patch.
4. **Verify Deployment** - A patch’s deployment can be verified to ensure that it has been installed successfully and taken effect. The robustness of verification can vary a great deal and is largely dependent on an organization’s needs, but automated means are generally needed to achieve verification at scale.
5. **Monitor the Deployed Patches** In the last phase of the life cycle, the patch’s deployment can be monitored using automation to confirm that the patch is still installed.

This document shall be reviewed annually or as needed.

Document Source Reference #	NIST SP 800-40, Version 4, *Guide to Enterprise Patch Management Planning: Preventive Maintenance for Technology* (Apr. 2022)

Compliance Sources

Name	National Institute of Standards and Technology (NIST), Computer Security Resource Center (CSRC)	*Website*	http://csrc.nist.gov/
Contact Information	inquiries@nist.gov		
Name		*Website*	
Contact Information			

Keywords

List Keywords	System Life Cycle, exploitation, bugs, vulnerabilities, malicious, remediation, maintenance, monitor, threats, holes, defects, bad code, Patch and Vulnerability Group, PVG.

Component Classification

Provide the Classification	□ *Emerging*	☒ *Current*	□ *Twilight*	□ *Sunset*
Sunset Date				

Component Sub-Classification

Sub-Classification	Date	Additional Sub-Classification Information
□ *Technology Watch*		
□ *Variance*		
□ *Conditional Use*		

Rationale for Component Classification

Document the Rationale for Component Classification	

Migration Strategy

Document the Migration Strategy	

Impact Position Statement

Document the Position Statement on Impact	

Current Status

Provide the Current Status	□ *In Development*	☒ *Under Review*	□ *Approved*	□ *Rejected*

Audit Trail

Creation Date	07-06-06	*Date Approved / Rejected*	02/19/2025
Reason for Rejection			
Last Date Reviewed	02/19/2025	*Last Date Updated*	02/19/2025
Reason for Update	Vitality		