



COMPLIANCE COMPONENT

DEFINITION	
<i>Name</i>	Enterprise Patch Management
<i>Description</i>	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.
<i>Rationale</i>	Enterprise patch management reduces or eliminates the potential for exploitation.
<i>Benefits</i>	Potentially eliminates the exploitation of vulnerabilities.
ASSOCIATED ARCHITECTURE LEVELS	
<i>Specify the Domain Name</i>	Security
<i>Specify the Discipline Name</i>	Management Controls
<i>Specify the Technology Area Name</i>	System Life Cycle Security
<i>Specify the Product Component Name</i>	
COMPLIANCE COMPONENT TYPE	
<i>Document the Compliance Component Type</i>	Guideline
<i>Component Sub-type</i>	
COMPLIANCE DETAIL	
<i>State the Guideline, Standard or Legislation</i>	<p><i>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.</i></p> <ol style="list-style-type: none"> 1. Risk Responses - Patching is one of several ways to respond to risks from software vulnerabilities. <ol style="list-style-type: none"> 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.

	<p>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.</p> <ul style="list-style-type: none"> 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: <ul style="list-style-type: none"> 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. <p>3. Risk Response Execution – Below are the common phases of executing a risk response, specifically within the context of patching.</p> <ul style="list-style-type: none"> a. Prepare to Deploy the Patch: <ul style="list-style-type: none"> 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: <ul style="list-style-type: none"> 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. <p>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.</p> <p>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.</p>
<p><i>Document Source Reference #</i></p>	<p>NIST SP 800-40, Version 4</p>

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	<input type="checkbox"/> Emerging	<input checked="" type="checkbox"/> Current	<input type="checkbox"/> Twilight <input type="checkbox"/> Sunset
Sunset Date			
COMPONENT SUB-CLASSIFICATION			
Sub-Classification	Date	Additional Sub-Classification Information	
<input type="checkbox"/> Technology Watch			
<input type="checkbox"/> Variance			
<input type="checkbox"/> 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	<input type="checkbox"/> In Development	<input type="checkbox"/> Under Review	<input checked="" type="checkbox"/> Approved <input type="checkbox"/> Rejected
AUDIT TRAIL			
Creation Date	07-06-06	Date Approved / Rejected	09-12-2006
Reason for Rejection			
Last Date Reviewed	08-17-2022	Last Date Updated	09-30-2022
Reason for Update	Vitality		