b

IT127   
Patch Management Policy

|  |  |
| --- | --- |
| Status | Final |
| Version | 1.01 |
| Classification | Confidential |
| Owner | Razvan Anghelidi, Directory of IT |
| Address | 1705 Tech Avenue, Unit 3, Mississauga, ON, L4W 0A2, Canada |

*Statement of Confidentiality: This document and supporting materials contain confidential and proprietary business information of Signifi Solutions Inc. These materials may be printed or photocopied for use internally and must not be shared with other parties.*

[Document Control iii](#_Toc61981394)

[Audience 4](#_Toc61981395)

[Scope 4](#_Toc61981396)

[Policy Statement 4](#_Toc61981397)

[Scope 5](#_Toc61981398)

[Compliance 13](#_Toc61981399)

[Enforcement 13](#_Toc61981401)

[Update 13](#_Toc61981402)

[Revision History 13](#_Toc61981403)

All rights reserved

No part of this document may be reproduced in any form, including photocopying or transmission electronically to any computer, without prior written consent of Signifi Solutions Inc (from now on, Signifi). The information contained in this document is proprietary to Signifi and may not be used or disclosed except as expressly authorized in writing by Signifi.

Trademarks

Other product names mentioned in this document may be trademarks or registered trademarks of their respective companies and are hereby acknowledged.

Document Control

The electronic version of this document is recognized as the only valid version.

Approval History

|  |  |  |
| --- | --- | --- |
| APPROVER(S) | TITLE/DEPARTMENT | APPROVED DATE |
| Shamira Jaffer | CEO | December 23rd, 2021 |
| Shamira Jaffer | CEO | December 7th, 2020 |
| Shamira Jaffer | CEO | July 10th, 2020 |
|  |  |  |
|  |  |  |

## Document Sensitivity Level

Confidential

Audience

This policy covers all individuals working at all levels and grades, including senior managers, officers, directors, employees, consultants, contractors, trainees, homeworkers, part-time and fixed-term employees, casual and agency staff and volunteers (collectively referred to as staff in this policy).

Scope

All staff are expected to always comply with this policy to protect the privacy, confidentiality, and interests of our company and our services, employees, partners, and customers.

Policy Statement

All machines shall be regularly scanned for compliance and vulnerabilities. All vendor updates shall be assessed for criticality and applied at least monthly. Critical updates should be applied as quickly as they can be scheduled.

Vulnerability Remediation / Patch installation Timeframe Requirement:

|  |  |  |  |
| --- | --- | --- | --- |
| **Vulnerability Severity Rating** | **Cloud Hosted systems** | **Corporate externally exposed systems** | **Corporate internal system** |
| **Critical** | 8 days | 8 days | 14 days |
| **High** | 8 days | 8 days | 30 days |
| **Medium** | 30 days | 30 days | 30 days |
| **Low** | Risk assessment-based decision, but not later than 60 days | Risk assessment-based decision, but not later than 60 days | Risk assessment-based decision, but not later than 90 days |

Scope

All Signifi Information Technology assets.

This policy applies to all components of the information technology infrastructure deployed and managed by Signifi:

• Servers based on:

- Windows OS

- Linux OS

• Databases

• Storage devices

• Load Balancers

• Web application servers

• Firewalls, Wi-Fi routers/access point and other virtual and physical appliances.

• And all other software and hardware components of the deployed infrastructure.

Risks / Challenges of Patch Management

Without effective vulnerability and patch management there the is risk of systems unavailability. This can be caused by viruses and malware exploiting systems or by out-of-date software and drivers making systems unstable.

**Patch Scheduling**

Timing, prioritization, and testing are intertwined issues for enterprise patch management. Ideally, an organization would deploy every new patch immediately to minimize the time that systems are vulnerable to the associated software flaws.

DevOps and IT departments should schedule maintenance time frame windows for patching process, for all Signifi hosted on cloud platforms environments provided in Signifi’s SaaS portfolio.

Internal corporate IT department should schedule and publish their own time frame.

The window should be as close as possible and synchronized with official time frame of scheduled patch releases of products Signifi is depending on, for example Microsoft security updates are released on the second Tuesday of each month.

Alternative Host Architectures

Enterprise patch management is relatively straightforward when all the hosts are fully managed and running typical applications and operating systems on a regular platform. When alternative host architectures are employed, patch management can be considerably more challenging. Examples of these architectures include the following:

• **Unmanaged hosts**. As mentioned in the previous clause, it can be much more difficult to control patching when hosts are not centrally managed (i.e., users manage their own hosts, appliances, IoT devices). The Signifi kiosks are generally falling into this category.

• **Non-standard IT components** (e.g., appliances). On such hosts, it is often not possible to patch individual applications independently. Rather, the organization must wait for the component vendor to release updated software. This wait time may be significantly longer than that used by the primary application vendors, resulting in significant vulnerability windows.

Software Inventory Management

Enterprise patch management is dependent on having a current and complete inventory of the patchable software (applications and operating systems) installed on each host. This inventory should include not only which software is currently installed on each host, but also what version of each piece of software is installed. Without this information, the correct patches cannot be identified, acquired, and installed. This inventory information is also necessary for identifying older versions of installed software so that they can be brought up to date. A major benefit of updating older versions is that it reduces the number of software versions that need to be patched and have their patches tested.

Resource Overload

Enterprise patch management can cause resources to become overloaded. For example, many hosts might start downloading the same large patch (or bundle of patches) at the same time. This could consume excessive network bandwidth or, if the patches are coming from an organization patch server, overwhelm the resources of that server. Signifi should ensure that their enterprise patch management can avoid resource overload situations, such as by sizing the solution to meet expected volumes of requests, and staggering the delivery of patches so that the enterprise patch management system does not try to transfer patches to too many hosts at the same time.

Installation Side Effects

Installing a patch may cause side effects to occur. A common example is the installation inadvertently altering existing security configuration settings or adding new settings. This may create a new security problem in the process of fixing the original vulnerability via patching. Signifi should be capable of detecting side effects, such as changes to security configuration settings, caused by patch installation.

It is strongly recommended running vulnerability scanning after the systems were patched.

Patch Implementation Verification

As discussed in the previous clause, an installed patch might not take effect until the affected software is restarted or other state changes are made. It can be surprisingly difficult to examine a host and determine whether or not a particular patch has taken effect.

This is further complicated when there is no indication for a patch when it would take effect (reboot required/not required, etc.) One option is to attempt to exploit the vulnerability, but this is generally only feasible if an exploit already exists, and there are substantial risks with attempting exploitation, even under highly controlled conditions. Signifi should use other methods of confirming installation, such as a vulnerability scanner that is independent from the patch management system.

Patch management procedure (phases/steps)

1. **Discover**

The first step in Patch Management is to define your starting point. The first step is to identify and categorize your assets: taking a full inventory of all workstations and servers on your network.

Inventory report should involve the list of assets with OS version and application installed on it. Once your assets are identified, they need to be categorized based on exposure and risk.

By categorizing assets, you develop a picture of which machines require rapid patch management (within hours or days) and which require standard management (weeks). Categorizing your assets is almost always a manual process. It is difficult to automate a process that essentially identifies “important machines” and “less-important machines.” One consideration when categorizing machines is the information that machine protects. Other issues to consider are public visibility (as in the case of a website) and sensitivity (customer credit card numbers, PII, PHI). The most important question to ask is “what the impact on the company will be if this machine is down or compromised?”

***Risk Analysis should be an integral part of the Patch Management process.***

After getting all the information you should be proactively enroll to all security bulletins distribution list of all application and OS vendors in your inventory list, in order to get the updates/patches release notifications in real time.

1. **Analyze**

The next step is the analysis phase, in which current patch levels are assessed. It can be done by vulnerability or patch management systems (e.g. Nessus) that are designed to scan the systems they discover for installed and missing patches. The accuracy of this step is critical.

Worst case scenarios are false positives; reporting a patch as present when in fact it is not. This may result in the patch never being applied. The less-critical counter to this is a false-negative; reporting a needed patch is not present when in fact it is. This will usually result in the re-application of the patch, with little harm done.

This patch analysis is based on several different information points. Typically, the operating system needs to be determined for a given device, as well as which applications installed on the machine. Based on that information, most tools will consult a “master list” of patches that are available for a given OS and application and determine which of these patches are installed and which are not. This “Master List” is analogous to antivirus software virus definition files and should be downloaded regularly from vendor websites.

**Patch Testing**

Let us consider the results of the first two steps. You should now have a clear picture of your current patch levels. Patch Level Minimum Baselines an important concept is the minimum patch level you require on your network. This minimum patch baseline will be unique to each network and can only be determined by a thorough understanding of the analysis, research, and test phases.

1. **Research**

Before you begin the process of deploying any service packs or patches to your environment, it is STRONGLY recommended that you research what you are about to deploy. Occasionally, the application of a patch, or even service pack, can have an unexpected negative impact on a machine; therefore, it is necessary to understand what you are deploying to your environment. Review resources such as the MITRE and NIST websites, where vulnerabilities are reviewed and detailed. Vendors publish articles describing vulnerabilities and include release notes and/or a read-me files describing installation options and precautions. Vendor testing should never replace your own, however. Every environment is different and third-party or custom software makes interactions unique and unpredictable. Based on the information you collect; you should determine the following for each patch you deploy:

• What is the nature of the vulnerability?

• What application or OS component is affected by it?

• How easy is it to exploit the vulnerability?

• What is the severity of the vulnerability?

• If the vulnerability is exploited, how much damage could be caused?

• Vulnerabilities are typically rated as low, medium, high or critical, critical being the highest level of potential damage should the vulnerability go un-patched.

• What is your level of exposure to the vulnerability?

Use the above information to guide your deployment of patches. Conduct a risk analysis. For example, if you find a high occurrence of missing patches for severe vulnerabilities, you may wish to address those systems first. Is this a severe vulnerability on your mission-critical application servers, or is it a low-severity across internal workstations? Based on that determination, you can begin to address the issues of testing the new patches for deployment.

***A Few Precautions***

It should also be noted that, in the case of major system upgrades (and some small ones, too), reasonable precautions should be taken before making any change. This includes reading release notes and any deployment guide. There may be recommendations to back up critical data or the entire system before deployment, so read carefully.

1. **Test**

The reason for testing patches prior to deployment may be obvious but should be stated clearly: patches sometimes break operating systems or application. It is a fact of life in patch management. Even in the case of a fully tested service pack, there is always a chance that it will conflict with some as-yet-undiscovered quirk in a small number of environments and, when that conflict occurs, servers go down. Therefore, the importance of testing in your own environment, on your own machines, cannot be stressed enough.

The testing phase of deployment includes applying patches to a test environment prior to deploying them to a production system. Sometimes the nature of a patch is that it has been written quickly to address a critical issue. Therefore, there is not always time to thoroughly test a patch prior to release. This is not to imply that patches are untested; but the testing is not nearly as extensive as in the case of a service pack, which goes through beta testing and review prior to release.

Of course, service packs should not be immune to the testing phase. Although they are tested thoroughly by their vendors, no vendor can test every update in every possible environment, so no patch or service pack should ever be deployed without being tested in your own test environment first.

***So how do you test a patch?***

Deploy it to a test/staging machine configured like the production system(s) that need the patch. It is strongly recommended that you provision a test environment and use that environment to test patch deployment before deploying to production.

Whatever your test environment is, deploy patches run acceptance tests to that environment. Then observe and record the results:

• Is the system still functioning?

• Are the applications and services on it still functioning?

• Do the results of the install coincide with the expected results:

* Are application extensions updated; are registry keys changed?
* If no negative impact is determined, the patch can be deemed safe
* If a problem occurs, go back to the research phase.

1. **Deploy**

• ***Create a backup/snapshot*** and verify its integrity by deploying it on a standby system.

• If the system is critical, and deployed in a high-availability configuration with component redundancy behind a Load Balancer (LB), prepare the LB, remove the system from production destination pool of LB.

• Patch the system. Validate base functionality of the application.

• Swap the patched standby system into production and keep the previous unpatched production system as a standby for emergency patch regression.

• Closely monitor the patched production system for any issues not identified during testing.

• Patch the standby system (old production) after confidence is established with the production unit.

• Update configuration management task and related records (e.g. QuickBase ticket).

1. **Report**

Reporting is the final step in the Patch Management process. You must be able to confirm the successful deployment of patches and verify that there is no negative impact. Reporting should expose situations that require an immediate return to the analysis phase, such as a failure in deployment. Reporting also allows an opportunity to review patch management process and look for areas of improvement, as well as providing valuable statistical information regarding patching activity. In environments where internal or external audits (often to meet industry or regulatory regulations) are required, documentation of changes is crucial to proving compliance.

Reporting conducts a change review and verifies successful deployment of patches. Reporting should also include enough review, analysis, and adjustment to close the loop and begin the cycle again automatically.

Compliance

This policy will be officially monitored for compliance by the IT department director and may include random and scheduled inspections.

Enforcement

All instances of non-compliance will be reviewed by the employee’s department director. The department director, with the assistance of the Human Resources department has the authority to impose disciplinary actions, up to and including termination of employment or contractual agreement.

Update

This policy and all supporting documentation will be reviewed and updated annually or upon material changes to Signifi business rules, technology processes, organizational goals, or information security objectives to ensure its continuing suitability, adequacy, and effectiveness.

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| VERSION | DATE | SUMMARY OF CHANGE | CHANGED BY |
| 1.0 | 2020-06-22 | First version | Muhammad Nasir |
| 1.02 | 2020-12-02 | Annual review | Razvan Anghelidi |
| 1.03 | 2021-12-12 | Annual review | Hadeel Alzuhairi |