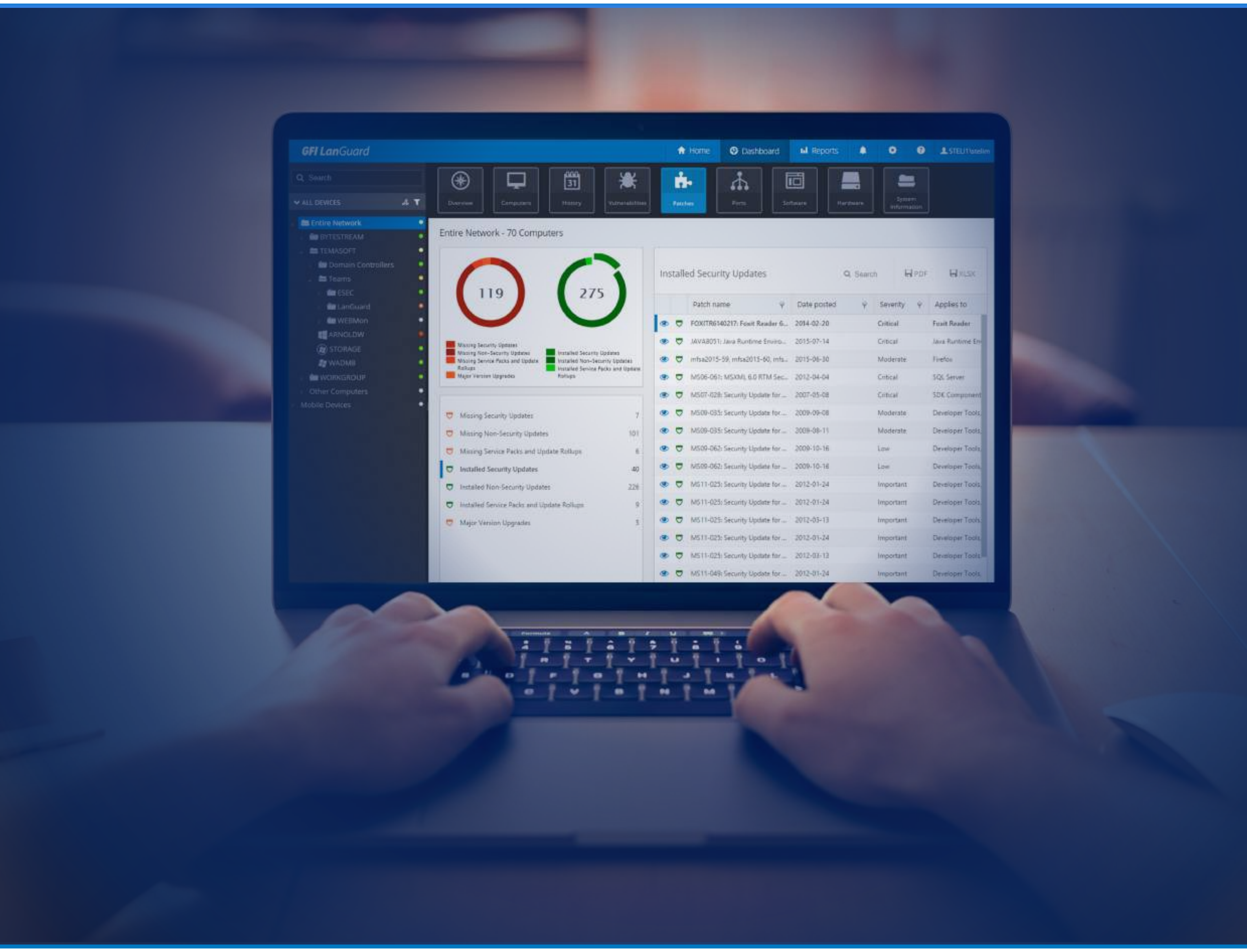
WHITEPAPER

6-Step Guide to Effective Patch Management



GFI Software™

Table of Contents

Introduction

Effective network patching is a key part of every cybersecurity strategy. Patching is also frequently a compliance requirement. Many organizations are required to have a patch management system in place to maintain compliance with regulations and standards such as ISO 27001 and ISO 270012.

We expect patch management to reduce vulnerabilities, improve performance, improve usability and assist in achieving compliance. However, this is not always the only outcome. One vulnerability may be ‘patched’ yet we are left with other problems. Patches may cause conflict with other software and hardware within our system environment. They may create new network problems that were not present before the patching, resulting in user dissatisfaction and frustration.

Patch management should be proactive, strategic and a planned process to determine the application of patches needed to specified systems at a specified time. Without an optimal patch management system in place, organizations are not effectively managing security and risk and may be inadvertently adding to their stockpile of “things to do.”

Challenges to effective patch management

- Patch volume...the volume of security patches is enormous with thousands of products with potential vulnerabilities.
- Increasing amounts of applications and utilities increases the “surface area” for attack.
- Lack of standardization and accreditation within the organization for systems.
- Not having a “Policy for Patching” in place, demarcating the procedures, roles and responsibilities.
- Tracking relevant patches and maintaining a risk assessment.
- Lack of available resources and time.
- Increases in mobile computing.
- Insufficient reporting on risk areas as collecting data proves difficult.
- Poor software management, variations in deployed versions of software.
- No automation in place for maintaining patch management in a consistent manner.

Benefits of an effective patch management system

- ✓ **Increased employee productivity -** Reduced downtime from malware issues.
- ✓ **Security -** Lower rates of virus infections, malicious attacks, and data theft or loss.
- ✓ **Compliance -** An automated patch management system can assist in keeping your environment patched at all times. Failure to comply can mean serious consequences for many organizations from legal and financial penalties or even closure.
- ✓ **Cost savings -** Fewer resources spent on fixing devices, as patches can solve inherent problems.
- ✓ **Increased IT productivity -** Manual patching requires significant IT resources and time. Automation means IT resources can be used elsewhere.
- ✓ **Additional capabilities -** Patches may extend software features and functionality or additional support.

6 steps to achieving an effective patch management system

1 Establish a patch policy for your organization

Effective policies include the following:

- Scope that describes what should be patched (determined through data type, asset value, location and organization objectives).
- Agreed timing for when updates or patches should be applied.
- A patch exclusion procedure and who is responsible for authorising this. The exclusions should be tractable at all times.
- Up-to-date and maintained Asset Inventory Management. Automated scanning of installed programs and binary files will help assess where patching is necessary.

2 Assign responsibility for identifying and distributing patches within the organization

This team will be responsible for:

- Managing risk through patching procedures.
- Keeping an inventory of company resources: identifying hardware, operating systems, and applications in use within the organization that are in need of patching.
- Assessing vulnerability, risk and impact to prioritise corrective measures in a planned manner.
- Ensuring procedures exist for corrective measures and that they are maintained and can be applied through the organization.
- Testing the patch in a controlled environment to ensure that the patch is not a cause of conflict with other applications within your organization before full deployment. It's important to rule out conflicting behaviour to avoid frustration at a later stage.
- Conducting a risk assessment associated with deploying the patches.
- Performing automated deployment of patches with the necessary tools and configuring automated updates wherever possible and suitable.

3 Verify patch installation and failure resolution

It's important to verify that the patch is installed correctly after deployment. If the patch fails to install or is installed incorrectly, you require a resolution procedure to be in place and followed. Effective patch management includes verification to ensure that the patch is present after installation.

A help desk for end-user support associated with patching can help with this step.

4 Use automation where possible and suitable

Automation is the route to sustainable patch management. Manual patching is not a viable solution for the long-term. For effective, automated patch management, take care to manage the tools and settings so you don't open up to further risks through their use.

It's also recommend to apply patches in phases.

5 Create a database for corrective actions

The database tracks corrective measures and patch exclusions. Managing patches and exclusions over time becomes challenging if you do not have some form of risk database. It helps you maintain control and track the corrective actions that need to be applied.

6 Review the effectiveness of your patch management system

Validate the effectiveness of your patch management system and have visibility of the current vulnerability state of the organization's systems. By gauging criteria such as:

- the maturity of the patch management system
- cost involved to deploy the patch management system
- compliance and risk

the performance of the system can be measured.

This should be a continuous process and you should consider necessary changes if and when needed

Summary

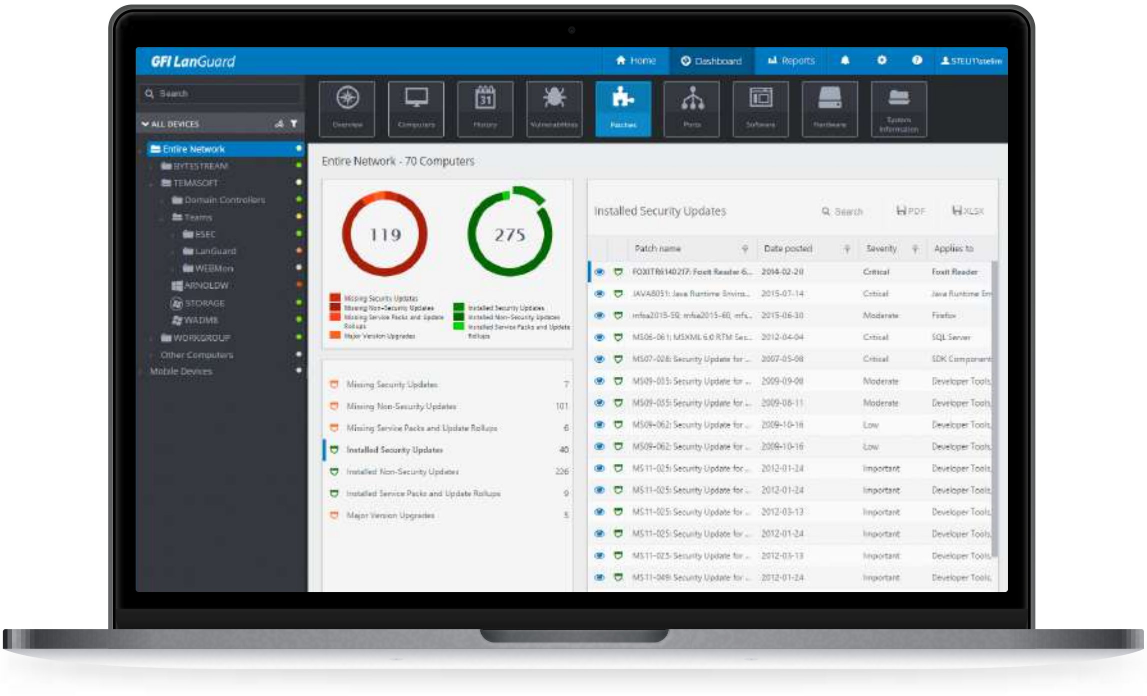
Without an automated patch management system in place, the likelihood of maintaining effective patching is greatly reduced. A patch management system ensures you maintain security while using less organization resources than manual management.

The importance of testing patches before patch deployment cannot be emphasized enough. Patches can break things, cause conflicts, and create problems with other software. Test environments should be mandatory.

Many people believe that the Microsoft recommended patches cover the majority of vulnerabilities. However, this is not the case. On the contrary, these vulnerabilities are only a fraction of those you are likely to face on a daily basis. You cannot solely rely on the patches, updates and service packs supplied through Microsoft and assume all vulnerabilities are covered. Following this approach without thinking twice or testing prior to patching may be the cause of unnecessary frustration brought about by avoidable conflicts and breakages. Test before you patch.

Many applications exist outside of the operating system and they contribute to your large surface area of vulnerability.

Vulnerabilities in software will continue to be a risk factor for your organization. An effective patch management system and process is essential.



Entire Network - 70 Computers

Missing Security Updates	7
Missing Non-Security Updates	101
Missing Service Packs and Update Rollups	6
Installed Security Updates	40
Installed Non-Security Updates	226
Installed Service Packs and Update Rollups	9
Major Version Upgrades	5

Installed Security Updates

Patch name	Date posted	Severity	Applies to
FOXITR6140217: Foxit Reader 6...	2014-02-20	Critical	Foxit Reader
JAVA8051: Java Runtime Enviro...	2015-07-14	Critical	Java Runtime En
mfsa2015-59; mfsa2015-60; mfs...	2015-06-30	Moderate	Firefox
MS06-061: MSXML 6.0 RTM Sec...	2012-04-04	Critical	SQL Server
MS07-028: Security Update for ...	2007-05-08	Critical	SDK Components
MS09-035: Security Update for ...	2009-09-08	Moderate	Developer Tools
MS09-035: Security Update for ...	2009-08-11	Moderate	Developer Tools
MS09-062: Security Update for ...	2009-10-16	Low	Developer Tools
MS09-062: Security Update for ...	2009-10-16	Low	Developer Tools
MS11-025: Security Update for ...	2012-01-24	Important	Developer Tools
MS11-025: Security Update for ...	2012-01-24	Important	Developer Tools
MS11-025: Security Update for ...	2012-03-13	Important	Developer Tools
MS11-025: Security Update for ...	2012-01-24	Important	Developer Tools
MS11-025: Security Update for ...	2012-03-13	Important	Developer Tools
MS11-049: Security Update for ...	2012-01-24	Important	Developer Tools

Get your **FREE** LanGuard trial

GFI Software™

All product names and companies mentioned may be trademarks or registered trademarks of their respective owners. All information in this document was valid to the best of our knowledge at the time of its publication. The information contained in this document may be changed without prior notice.