NISCC
NATIONAL INFRASTRUCTURE SECURITY CO-ORDINATION CENTRE

Good Practice Guide
Patch Management

Issued 24 October 2006

Abstract

This paper provides a guide to Critical National Infrastructure organisations on Patch Management. It describes a four-stage process for ensuring that all systems are appropriately patched to minimise security vulnerabilities, and describes a system of metrics that organisations may use for measuring the effectiveness of their patching strategy. Finally, it gives some specific details on patching tools for common operating systems.

Disclaimer

Reference to any specific commercial product, process or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement, recommendation, or favouring by NISCC. The views and opinions of authors expressed within this document shall not be used for advertising or product endorsement purposes.

To the fullest extent permitted by law, NISCC accepts no liability for any loss or damage (whether direct, indirect or consequential and including, but not limited to, loss of profits or anticipated profits, loss of data, business or goodwill) incurred by any person and howsoever caused arising from or connected with any error or omission in this document or from any person acting, omitting to act or refraining from acting upon, or otherwise using, the information contained in this document or its references. You should make your own judgement as regards use of this document and seek independent professional advice on your particular circumstances.

National Infrastructure
Security Co-Ordination Centre
PO Box 832
London
SW1P 1BG

Tel: 0870 487 0748
Fax: 0870 487 0749
Email:enquiries@niscc.gov.uk
Web: www.niscc.gov.uk

Contents

1 Key Points

- This document is a guide to *patch management*, defined as the process of controlling the deployment and maintenance of interim software releases into operational environments.

- If an organisation cannot determine and maintain a known level of trust within its operating systems and application software, it might have a number of security vulnerabilities, which, if exploited, could lead to significant business impact. Properly configuring IT systems, using up-to-date software, and installing recommended software updates will all help to mitigate the risks of such exploitation.

- This guide proposes a four stage patch management process.

    - **Assessment and Inventory** The purpose of this phase is to accurately record what software components comprise the operational environment, what security threats and vulnerabilities exist, and whether an organisation is prepared to respond to new software updates.

    - **Patch Identification** The purpose of this phase is to identify patches and software updates as they are released, determine whether they are relevant to the organisation, and determine whether an update represents a normal or emergency change.

    - **Evaluation, Planning and Testing** The purpose of this phase is to decide, for any given patch, whether to deploy that patch into the operational environment, to plan how and when that deployment will take place, and to test the software update in a realistic operational environment to confirm that it does not compromise business critical systems and applications.

    - **Deployment** The purpose of this phase is to successfully roll out the approved software update into the operational environment whilst minimising impact on system users.

- The document also suggests metrics by which an organisation might assess the effectiveness of its patching strategy, including metrics to.

    - Measure a system's susceptibility to attack.

    - Measure mitigation response time.

    - Measure the cost of patch and vulnerability management.

2 Patch Management Overview

Patch management is the process of controlling the deployment and maintenance of interim software releases into operational environments. It assists in maintaining operational efficiency and effectiveness, mitigating security vulnerabilities, and maintaining the stability of an organisation's production environment.

If an organisation cannot determine and maintain a known level of trust within its operating systems and application software, it might have a number of security vulnerabilities, which, if exploited, could lead to significant business impact. Properly configuring IT systems, using up-to-date software, and installing recommended software updates will all help to mitigate the risks of such exploitation.

To consider the full business impact of poor or non-existent patch management, consider the following.

- **Downtime** What is the cost of computer downtime in your organisation? What if business or nationally critical systems are interrupted? Consider also the opportunity cost of lost end-user productivity, missing transactions on critical systems, and lost business during an incident. Most hacking attacks result in some downtime, either as a direct result of the attack itself or as a necessary part of remediation efforts. Some attacks have left computers down for several days.
- **Remediation time** What is the cost of fixing a wide-ranging problem in your organisation? How much does it cost to rebuild a computer's software environment? Many security breaches require a complete reinstallation to be certain that back doors (permitting future exploits) are not left by the attack.
- **Questionable data integrity** In the event that an attack damages data integrity, what is the cost of recovering that data from the last known good backup, or confirming data correctness with customers and partners?
- **Lost credibility** What is the cost of lost credibility with your customers? How much does it cost if you lose one or more customers?
- **Negative public relations** What is the impact to your organisation from negative public relations? What would be the effect on your business if you are seen as an unreliable organisation with which to do business, or if you fail to protect your customers' personal information?
- **Legal defences** What might it cost to defend your organisation from others taking legal action after an attack? Organisations providing important services to others have had their patch management process (or lack of one) put on trial.
- **Stolen intellectual property** What is the cost if any of your organisation's intellectual property is stolen or destroyed?

In addition to the business impact of poor patch management, the impact for those information systems which have been determined to be nationally critical also needs to be considered. The failure or compromise of a nationally critical system will not only affect your business, but will also adversely affect the social or economic well-being of the nation as a whole. Any changes to nationally critical systems need to be managed, and the exact impact of changes known before those changes are made.

Assessing and maintaining the integrity of software in a networked environment through a well-defined patch management program is the key first step toward successful information security, regardless of any restrictions to physical access to a computer.

3 A Recommended Patch Management Process[1]

Any patch management process must give the host organisation control over how and when interim software releases are deployed into the operational environment. This document recommends a four phase process, as follows.

- **Assessment and Inventory** The purpose of this phase is to accurately record what software components comprise the operational environment, what security threats and vulnerabilities exist, and whether an organisation is prepared to respond to new software updates.

- **Patch Identification** The purpose of this phase is to identify patches and software updates as they are released, determine whether they are relevant to the organisation, and determine whether an update represents a normal or emergency change.

- **Evaluation, Planning and Testing** The purpose of this phase is to decide, for any given patch, whether to deploy that patch into the operational environment, to plan how and when that deployment will take place, and to test the software update in a realistic operational environment to confirm that it does not compromise business critical systems and applications.

- **Deployment** The purpose of this phase is to successfully roll out the approved software update into the operational environment whilst minimising impact on system users.

Figure 1 - illustrates the process and its four phases.

[1] This process is largely based on that described in the Microsoft patch management guide [2], which may be consulted for further information.

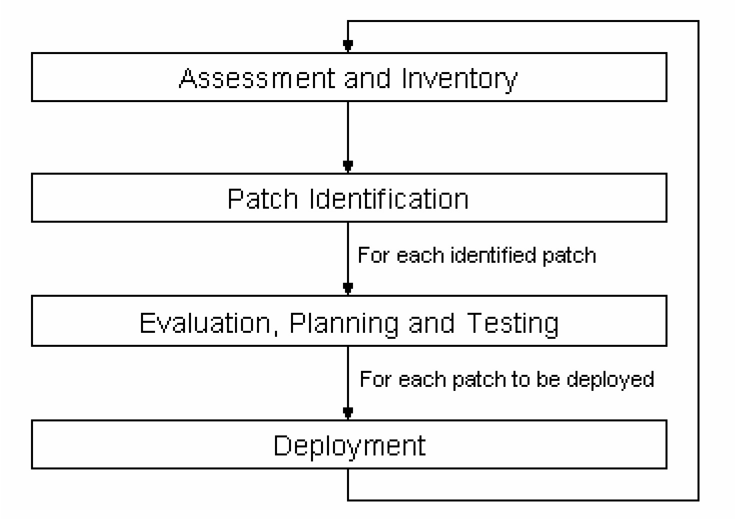
Assessment and Inventory
Patch Identification
For each identified patch
Evaluation, Planning and Testing
For each patch to be deployed
Deployment

Figure 1 - Four Stage Patch Management Process

It is important to stress that the process of patch management need not be a heavyweight process. Organisations are encouraged to adapt the process to their own business needs.

The sections below explain each of these four stages in detail.

3.1 The Assessment and Inventory Phase

The assessment and inventory phase is an ongoing process to determine what computing assets are present in an organisation, how they can be protected, and how the organisation can develop a software distribution architecture to support patch management.

The phase contains the following individual steps.

- Inventory existing IT systems.
- Apply baseline system configurations.
- Identify security policies and technical standards.
- Set up or review ongoing security processes.
- Determine the best source for information about new software updates.
- Assess the existing software distribution infrastructure.
- Assess operational effectiveness.

3.1.1 Inventory Existing IT Systems

Effective patch management requires accurate and current knowledge of what hardware and software has been deployed in the operational environment. Without this information, it will be impossible to accurately determine which computers in the organisation require a software update. It is also important to determine the manner in which client computers are connected to the network. If some connect through slow or unreliable links or use remote access, such as dial-up facilities, the method of distributing and installing a software update will differ from the method used for computers that are connected to a fast, reliable network.

As a minimum, the following information must be retrieved from computers deployed in the operational environment in order to perform effective patch management.

- **Hardware Types and Versions** Understanding whether a computer is a portable computer (mobile client), desktop computer, or server will help administrators to determine how a particular software update needs to be installed. When patching a server, for example, it may be necessary to observe outage windows (specific times at which changes and computer restarts are permitted) or ensure that the server is backed up before the software update is deployed.
- **Operating System Types and Versions** Administrators must be aware of all operating system types and versions that have been deployed into the production environment.
- **Applications and Middleware** As with the operating system and service pack version, administrators must be aware of all deployed software applications and versions.
- **Computer Role** Determining the function of individual computers is essential in order to evaluate the impact of restarting the computer once a software update has been deployed. For example, if the computer is a server running a business-critical application, it is advisable to schedule software update during periods when they will have minimal impact on the business. It also may be necessary to make arrangements for business continuity, for example, so that users can continue to make use of the application while the server is being restarted.
- **Network Architecture and Connectivity** Understanding the layout of the network infrastructure, its capabilities, security level, link speed, and link availability is important for effective patching. Software updates can vary in size, and knowing the constraints of the network infrastructure can potentially reduce any delays in distributing software updates. It can also dictate the manner in which the software update will be deployed to particular client computers.
- **Installed and Missing Software Updates** Identifying which software updates have or have not been installed on computers is essential.
- **Legacy status of the computer system** If software or hardware updates are not available for a computer system this needs to be recorded, because the system will fall outside the patch management regime and will need a separate security management regime, such as a hardened configuration or use of multiple layers of defence (defence-in-depth).

An audit helps an organisation to understand and gain an accurate record of its operational environment, and hence to determine a baseline against which patch management can be carried out.

Once the audit has been carried out, administrators need to check the results for completeness and ensure that all managed computers have reported up-to-date information. The following activities should be performed.

- The audit results should be compared with the last (i.e. immediately previous) audit, and an increase or decrease in computers noted. A change in the count as compared to prior audits may indicate a gap in completeness (unless the discrepancy is accounted for by the number of new computers introduced into the system minus the number of computers retired).
- Systems in active use will register and use name resolution services such as Domain Name System (DNS). Scripts that cross-reference against active infrastructure such as these services may give a comprehensive view of how complete the inventory for patch management is. Ensure that scavenging or cleanup of stale records is enabled on any name resolution servers to obtain more accurate results.
- For IT systems using Microsoft Active Directory (AD), the audit results can be compared against computer objects (computer accounts) in the AD directory service domains. As long as stale accounts are cleaned up, this can be as useful as comparing against name resolution services.

If analysis discovers a discrepancy in the information returned you should investigate further to determine the cause and take steps to ensure that systems missed by the audit are included in the next audit run.

3.1.2 Applying baseline system configurations

In the context of patch management, a *baseline* is a set of documented configurations of a product or system that is established at a specific point in time. Baselines establish a standard that systems of the same class and category must match. Effective IT operations use baselines as a trusted point from which systems are built and deployed. Typically, the configuration defined by a baseline is stringently tested and vendor-certified.

An application or software baseline provides the information required to rebuild a system to a desired state. It might be necessary to establish baselines for different software applications, hardware vendors, or types of computers.

Baselining requires an accurate and up to date inventory of the computers and services in the IT environment. Keep in mind the following points when establishing baselines for an environment.

- Infrastructure that falls below the baseline must be addressed through problem management, in order to bring all the computers below the baseline up to compliance. These computers may have had issues with distribution, schedules, or permissions, or may require special care through exception handling.
- Infrastructure that exceeds the baseline is not necessarily at an advantage. Computers exceeding their class baseline (i.e. the baseline defined for all

systems in a given category or configuration) should be checked to determine if unauthorised changes have occurred. In some cases, it may be necessary to return a system to a trusted level or it may be appropriate to control it through a change freeze. Systems that exceed an approved baseline may contain application versions or software updates that have not been tested for interoperability and formally approved.

- Some systems may have special circumstances that make them exempt from the baseline of their class. For example, an older workstation running a legacy payroll application that connects to a processing agency by means of a modem may require an operating system level far below the established baseline. It may not be appropriate to upgrade this system to the latest baseline since this could prevent the legacy application from running.
- The most effective patch is the one you don't have to apply. In other words, when creating a baseline, consider disabling any services that are not supported by a business requirement. Even if there are vulnerabilities in those services, they cannot be exploited if the service isn't running.

3.1.3 Identify Security Policies and Technical Standards

Because patch management is part of IT risk management, it is recommended that all organisations using this guide also follow UK government guidance on the risk management and accreditation of IT systems [3], or equivalent guidance.

Regardless of the policy framework followed, the patch management process requires all security policies and technical standards used in the organisation to be recorded. This documentation may include.

- Installation standards, describing supported installation locations and methods.
- Network and domain standards, indicating how names and Transmission Control Protocol/Internet Protocol (TCP/IP) information is assigned and which domains computers should join.
- Operating system security options and policy settings, including those for reducing open ports based on required services.
- Any standards that describe the use of encrypted file systems.
- Minimum service pack or software update compliance, updated with each security release.
- Antivirus software compliance.
- Application security configuration settings, such as macro file protection and security zones.
- Administrative account standards, such as renaming or disabling accounts and setting up decoy accounts.
- Password standards.

Effective security policies should identify the minimum security standards for computers and help to minimise exposure to potential security vulnerabilities. To be effective, an organisation's security policies should be reviewed whenever changes are made to IT systems and software in the production environment. A security policy violation suggests a vulnerability that should be eliminated from the environment. The vulnerability could be a computer that lacks various software updates or is not configured correctly, or a user who fails to use strong passwords.

Security policies should provide guidelines for each type of policy-compliance violation; this can then be used to determine the severity of an incidence of a specific vulnerability.

Strong management structures are needed to effectively enforce security policies. The following practices support effective management.

- Security policies, enforcement timelines, and approaches must have the support of management across the entire organisation.
- Techniques and tools for determining ownership and administrative information on an unmanaged computer should be available to specific service desk technicians.
- Service desk technicians should be trained on mitigating each of the vulnerabilities that violate security policy.
- If possible, automated tools and techniques should be used to ensure that computer systems are continually in compliance with corporate security policy and standards.

The nature of a security vulnerability, the likelihood of exploitation and the cost of recovery should help determine how aggressive the response should be to a system that is not in compliance. If attempts to resolve the vulnerability in the required timeline are unsuccessful, it may be necessary to remove the computer from the network by physically disconnecting it (or configuring network hardware to automatically remove the device from the network) until remediation is successful. For further details see the NISCC First Responders Guide: Policy and Principles [7].

3.1.4 Set up or review ongoing security processes

Ongoing scanning and reporting of security issues is crucial for ensuring that potential security vulnerabilities are identified and addressed. At a minimum, organisations should perform the following actions.

- Regularly scan all computers for virus infection.
- Deploy vulnerability scanning tools to identify vulnerabilities in computer systems and infrastructure devices. See NISCC Technical Note 08/04, “Introduction to Vulnerability Assessment Tools” [8] for details of the types of scanners available.
- Review information returned by network monitoring tools, event logs, and other monitoring tools, and the output of any intrusion detection system to

determine whether attacks are being made against computer systems in the production environment.

- Create and maintain operating system images that can be applied to a given hardware platform to quickly revert a compromised system to a known good operating system.
- Check that all users and administrators are aware of the steps they need to take in the event of an attack on computer systems in the production environment.
- Maintain a prioritised list of all key information and assets that need to be protected first should an attack occur.
- Verify router and firewall logs and configurations to ensure they are consistent with your organisation's given standards for these devices.
- Review domain controller security policies.

Scanning systems for potential security vulnerabilities should be as automated as possible and performed on a regular basis: daily for most systems. The frequency will depend on the level of automation that is available for each of these areas, the availability and skills of IT security staff in the organisation, and the level of commitment to a secure environment.

3.1.5 Determine the Best Source for Information about Software Updates

Sources of information about software updates include.

- Computer Emergency Response Teams (CERTs)
- Independent web sites
- Vendor web sites
- E-mail notifications.

Subscribing to the proper notification methods is essential to maintaining and updating established operating baselines and for implementing an efficient patch management process.

CERTs allow the communities served by those CERTs to find out about vulnerabilities and software updates. Uniras is the CERT for UK government and the UK Critical National Infrastructure. For further details see http://www.niscc.gov.uk/niscc/respToIncidents-en.html.

Independent web sites, such as the NISCC web site at http://www.niscc.gov.uk/niscc/index-en.html, publish alerts on newly-discovered vulnerabilities and their corresponding patches as soon as they become available. Critical National Infrastructure organisations can also subscribe to receive the NISCC Uniras alerts and briefings by emailing uniras@niscc.gov.uk, specifying the exact email address to be subscribed.

Alternatively, all major software vendors and suppliers have their own security web sites which will include details of vulnerabilities and patches. Vendors may also offer one or more e-mail advisory services, allowing subscribers to receive security updates directly.

It should be noted that security updates (other than emergency updates) tend to be released at scheduled times. For example, Microsoft publishes details of new updates in bulletins which are published monthly on the second Tuesday of the month between 10am and 11am Pacific Time (GMT -8 hours).

3.1.6 Assess the Existing Software Distribution Infrastructure

Assessing the software distribution infrastructure is another key part of an effective patch management process. The assessment should address such questions as.

- Is a software distribution infrastructure in place?
- Can it be used to distribute software updates?
- Does it service all computers in your environment?
- Is it designed to handle patching of business-critical computers?

3.1.7 Assess Operational Effectiveness

Effective IT operational processes are perhaps the most important dependency for effective patch management. There are several questions to ask when assessing operational effectiveness.

- Are there enough skilled people to perform patch management?
- Are people aware that patch management is necessary?
- Do those responsible understand security settings, common computer vulnerabilities, software distribution techniques, remote administration, and the patch management process?
- Are there standard operational processes in place, or are day-to-day operations largely unstated and imprecise?
- Do processes exist for change management and release management, even informal ones?

An organisation should also review the current administration model of its IT environment and determine how well this supports patch management. It is advisable to consider the following issues.

- How large is the infrastructure compared to the number of staff? What is the administrator-to-system ratio?
- What is the current support model: centralised, decentralised, or shared?
- What are the hours of operation for support staff? Are there sufficient change windows allotted for maintenance and administration?

- Are roles clearly defined and communicated to team members?
- Are service management processes formalised and being followed?
- Are there sufficient tools for individuals to effectively execute their roles?

Finally, the following questions may help to determine the appropriate skill sets required for effective patch management execution.

- Do individuals have sufficient experience in handling the size and complexity of the infrastructure?
- Are personnel trained correctly and with relevant technologies?
- Are individuals working together well as a team?
- Do individuals have the appropriate experience or training to understand key operational disciplines and methodologies?
- Do individuals have skills in scripting?
- Do individuals understand user and system permissions and contexts?
- Do individuals understand software update structures and dependencies?

3.1.8 Summary

These are the key points to remember from the Assessment and Inventory phase of the patch management process.

- You should understand what your IT assets are, and which are business or nationally critical.
- You should understand what you have deployed in your production environment and what is classed as managed (by management tools) and what is not.
- You should ensure that your software distribution tools are configured, maintained, and able to support normal and emergency patch management.
- You should ensure that your personnel have assigned roles and responsibilities and that they know how to respond in an emergency-that is, how to deal with the software update and how to mitigate its impact.

3.2 The Patch Identification Phase

The goal for this phase of the patch management process is to.

- Discover new software updates in a reliable way.
- Determine whether software updates are relevant to your operational environment.
- Obtain and verify software update files.
- Determine whether the software update should be considered a normal change or an emergency, and submit a request for change (RFC) to deploy it.

3.2.1 Discovery of New Software Updates

Identification of a software update starts with discovering it in a safe and reliable way. Discovery has two main components.

- How notification of a new software update is received.
- How assurance can be gained that the notification is genuine.

Notification should be supplied either through subscription to a reliable source that provides scanning and reporting activities, or by some other reliable notification mechanism. The most commonly used notification mechanisms are e-mail notification and reports from vulnerability scanning tools.

It is important to handle e-mail notifications carefully.

- For security reasons, vendors typically do not attach software files to e-mail notifications. Instead, the e-mail is likely to point users to the vendor's security web site. It is advisable not to run or install any executable attached to an e-mail notification claiming to be from a software vendor.
- Do not click any links directly from inside an e-mail notification. Instead, retype the URL or copy the URL into a text only editor and then paste the text into a browser window in case the original URL displays a vendor's web site address but links to a malicious web site.
- Many vendors digitally sign all e-mail notifications related to security updates when it sends these notifications to customers. In such cases, always verify that the digital signature is valid before following the instructions in the notification.

3.2.2 Determine Whether Software Updates Are Relevant

Software updates are released with increasing frequency, from a variety of sources and for many reasons (not all of which are security related). The screening process outlined in this section should remove the majority of irrelevant software updates.

Each software update received should be checked for relevance. When a notification contains information about more than one software update, each update needs to be checked individually for its relevance to the organisation.

The first step in checking for relevance is determining whether the software update is designed to address the operating system or applications in your operational environment.
If so, check whether the application or system that the update applies to has the vulnerability the software update is designed to address.

Not every security update that applies to something in your environment will be relevant. Although it is important to be aware of, and have a good understanding of, existing security updates, you should only deploy security updates that have relevance to your environment. This will minimise the cost and effort required to keep your environment up-to-date and secure.

Although the information in a software update might be classified as irrelevant, it is important to record that it exists by passing the information to personnel in problem management and storing a copy of the update for dissemination if necessary. If it later becomes relevant and is required, the organisation will have access to this information at the original source.

Vendors typically rate security updates based on what access or level of privilege the vulnerability would allow the attacker should the vulnerability be exploited (for example, whether a vulnerability enables remote program execution). These ratings can be used to determine the urgency of any required actions.

Determining the relevance of technology-specific software updates can pose significant challenges in any environment. Technologies such as Microsoft’s Internet Information Services (IIS), which can be installed on clients or servers, can make it difficult to determine if a software update is relevant to any specific subset of computers in your environment. This emphasises the importance of maintaining as accurate an inventory of your environment as possible.

Each software update in a notification requires a detailed and in-depth review. This review should include all associated documentation, including that sent with the software update and supporting information that might be found, for example, on the vendor’s security web site.

Once you receive an e-mail message identifying applicable software updates, you must make someone responsible for investigating them. This team member should then take ownership of the software update.

The software update may be specific to particular scenarios or configurations. The reviewer should check whether the scenario/configuration deployed in production matches that covered by the notification or supporting web article.

There are also questions to ask in terms of software update dependencies.

- Are there dependencies relating to the update? For example, do certain features need to be enabled or disabled for the update to be effective?
- Does the software update require a certain service pack to be installed? Is the software update superseded by a service pack or another software update and does it makes sense to wait for a newer version?

Identifying the above dependencies is critical because it will have a direct impact on your release and deployment planning for the software update. Document which service pack the software update will appear in and whether a different version of the software update is required, depending upon the active service pack. It is important to know this in case compliance problems occur as users upgrade from one service pack to another.

3.2.3 Obtain and Verify Software Update Files

Once a software update has been identified and its relevance established, you must obtain the software update files from the supplier and confirm that they are safe and will install successfully. The verification process will either authenticate security updates or highlight updates that are not security-validated. In the latter case, when an invalid notification is received, information about the notification should be sent to those responsible for the subscription process and to the security team for further investigation. For example, if a notification comes from a normally reliable source of information, but the specific notification has errors on validation, this might raise security concerns about the quality of the notifications from this particular source. The source should be investigated and any issues resolved.

At a minimum, software update verification should consist of the following steps.

- Identifying the software update.
- Verifying the digital signature of the update (if available).
- Reviewing all accompanying documentation.
- Reviewing software update files.
- Identifying software update size.
- Identifying software update dependencies.
- Identifying any pre-patching or post-patching actions needed.
- Verifying that software update install procedures exist.
- Verifying that software update uninstall procedures exist.
- Ensuring that the software update is safe (by implementing it on a quarantined test network: see below).

To prevent virus infection or malicious code from affecting your IT infrastructure, all files related to a software update should be examined in an isolated (quarantined) environment. This quarantine should be imposed on all software and documentation. There should be strict controls in place in the quarantined environment and, to ensure this, the quarantine process should be carried out by a group of specialists in the organisation.

The review of relevant documentation should ideally be carried out by more than one person, to mitigate the risk of a single person missing critical and relevant points when evaluating the update. As you read the documentation, look for answers to the following questions.

- Will adopting the update cause other problems, resulting in a compromise of the production system?
- Do you need to perform any actions prior to deploying the update?
- Do you need to perform any actions after deploying the updates?
- Are there workarounds or mitigation steps available while you patch your environment?
- Are software update install procedures available?
- Are software update uninstall procedures available?
- What is the software update file size? File size will impact your overall release process and plan-for example, how you handle users who work from home or on the move.

On rare occasions, software updates may only require you to make registry or configuration file changes or adjust application settings, but most software updates will involve downloading files.

The following are some guidelines for verifying the software update install procedures.

- Determine whether the software update requires a restart. If it requires a restart, you will have to take into account special considerations during the planning and deployment phases for mission-critical or core infrastructure servers.
- Assess how much disk space the software update requires (including an Uninstall folder).
- Check whether the update provides configuration options that are available to you during the install.
- Read any supporting documentation for additional information about installing a software update.

Despite your testing, you may run into problems after installing the software update that require you to uninstall it. It is important, therefore, to test that the uninstall procedure works. After uninstalling, you should check that the server continues to run as expected and inspect the logs of any system monitoring tools.

3.2.4 Decide on the Nature of the Software Update and Submit an RFC

After identifying the software update, determining its relevance to the organisation, obtaining software update files, and confirming that they are safe and will install successfully, the next step is to submit a Request For Change (RFC) to start the evaluation and planning phase of the software update.

The change request submitted should address the following information.

- What is the change?
- What vulnerability is the change in response to?
- What services will be impacted by the change?
- Is a software update already being deployed for that service?
- Does the software update require a restart to complete the installation?
- Can the software update be uninstalled?
- What, if any, countermeasures can you implement to give you more time to test and deploy the software update?
- What are the recommended test strategies for this change?
- What is the suggested priority of the RFC?
- What is the impact (category) of the change?

If a software update addresses a critical security issue or system instability, the priority of the RFC should be marked as "emergency" An emergency RFC should be created only when the deployment of a software update or the implementation of security countermeasures such as closing network ports must be performed as a matter of urgency.

3.2.5 Summary

The following are the key points to remember from the Patch Identification phase.

- Ensure that you are notified of all new software updates.
- Confirm that the software update notification comes from an authorised source.
- Check that the software update is relevant to systems in your production environment.
- Obtain the software update files and confirm that they are free of viruses and other malware.
- Check that the software update installs successfully.
- Decide whether the software update is an emergency and submit an RFC to deploy it into production.

3.3 The Evaluation, Planning and Testing Phase

In this third phase, the software update is evaluated and - assuming that it is approved for deployment - its deployment into the production environment is planned and tested. By the end of the phase, the change request to deploy the update will have been reviewed and approved or declined. If approved, it will have been classed as either emergency or non-emergency.

The key requirements for this phase are to.

- Determine the appropriate response.
- Plan the release of the software update.
- Build the release.
- Test the release.

3.3.1 Determine the Appropriate Response

The software update RFC will describe the change required in the production environment, so others can act on it. The first step in the evaluation, planning and test phase is to review the RFC and determine the most appropriate response to a software vulnerability or threat. This will involve prioritising and categorising the request, then obtaining authorisation to deploy the software update.

3.3.1.1 Prioritising the RFC

Although priority and category are initially assigned by the change initiator and included in the RFC, those assignments have to be reviewed and either agreed to or changed before the change request can be authorized. The level of priority is particularly important because it determines how quickly a software update passes through the change process. The following considerations can help you to determine the priority level of a software update.

- What are the critical business (or national) assets? Will these be exposed to a potential security breach or system instability until the software update is installed? Change requests should be prioritised based on the impact of patching or not patching high-value assets.
- Will the software update apply to a system running a business-critical (or national critical) service, which has been the target of attackers in the past? This can be a good reason to raise the priority of a change request.
- Have you deployed countermeasures to reduce the exposure of a particular security vulnerability? This can lower the priority of the change request, although it may still be appropriate to deploy the software update to eliminate the vulnerability.
- What is the threat of the vulnerability in question to the operational environment? Many security bulletins and related software updates might

apply to only a few computers in your environment. If the threat of the vulnerability is low, that might lower the priority of the request.

- What level of access or privilege would the vulnerability allow an attacker if exploited? For example, would the vulnerability enable remote program execution, local privilege escalation or denial of service? Consider the technical impact in the context of the business or national criticality requirement.

The scale of priority levels is a matter for the organisation concerned. However, the following table details a four-point suggested priority scale, together with recommended time frames for implementing RFCs of each priority.

Category	Priority	Recommended Time Frame
Emergency	Emergency	Within 24 hours
Non-Emergency	High	Within 1 week
	Medium	Depending on availability, deploy a new service pack or update rollup that includes a fix for this vulnerability within 1-2 months.
	Low	Depending on availability, deploy a new service pack or update rollup that includes a fix for this vulnerability within 6 months.

Table 1 - Update Priorities and Recommended Deployment Timeframes

If high-value or high-exposure assets are impacted by the vulnerability, or the affected assets have been historically targeted by attackers, then there may be a case for raising the priority of the response from that originally calculated. Conversely, if extensive mitigating factors are in place, such as countermeasures that minimise the risk of compromise, or only low business impact assets are affected, then there may be a case for lowering the priority of the response from that originally calculated.

If the vulnerability that the security update addresses is already being exploited or is about to be exploited, or if the update corrects a system instability being encountered in the production environment, then there is a case for classifying the request as an “emergency”. This categorisation gives the request priority over all other changes happening within the operational environment.

3.3.1.2 Categorising the RFC

Whereas the priority of an RFC determines its urgency, the *category* of an RFC determines the amount of work required to implement it and the potential impact on the organisation’s IT systems during deployment. Correctly determining the category of an RFC is important, because it helps those reviewing the change to understand the impact it will have on systems and services within the production environment. To establish the category of the change request, it is necessary to determine.

- On which machines the software update needs to be installed and the roles (business criticality) of those machines. A software update that requires a business-critical computer to be rebooted, for example, will have a greater impact than one that does not.

- Whether additional changes will be needed to support the deployment of the software update. If, for example, the software update applies only to the current service pack, and that service pack is not installed on certain production systems, it may not be possible to protect those systems against a particular security vulnerability. In this case, the impact and hence the category of the change request would be greater, because both the service pack and the software update would need to be deployed.
- Whether the software update can be uninstalled once it has been installed. If not, then it presents a greater risk to the production environment than one that can be successfully removed. Although it may be necessary to deploy such software updates to provide protection from a particular security vulnerability or to address a particular system instability, the category of the request needs to reflect this.
- The likely impact on the network infrastructure. Deploying a large software update to many computers simultaneously could degrade network performance and adversely affect the proper operation of your entire environment. You should closely review all software update documentation, and always be aware of the software update's size and the number of computers that will receive it. This information can assist with properly scheduling the release.
- Whether certain services need to be stopped, paused, or closed during installation. This may affect an organisation's critical services or prevent an end user from working on the computer while the installation is taking place.

3.3.1.3 Obtaining Authorisation to Deploy the Software Update

Once the change request has been prioritized and categorized, it needs to be reviewed and authorised, before the software update can be deployed into production. To get the change request authorised, it is necessary to.

- Determine who should be involved in the decision-making process.
- Review the change request, assess the risks and consequences of deploying the software update, and select the most appropriate course of action.
- Identify who will be responsible for getting the software update deployed to all impacted systems.

It is important to determine who should be involved in reviewing and authorising a software update for deployment into production. Many organisations choose to set up Change Advisory Boards (CABs), made up of representatives from all areas of the business that will be affected. CAB members should include individuals who have experience in the specific technologies and services that will be used to deploy the update. Representatives from the business, network, security, service desk, and technical support teams may also be included in this group.

When a quick decision needs to be made, for example in the case of emergency software updates, authorisation may be delegated to a subset of the CAB, composed of people who possess the authority to approve emergency changes and who will be available to make quick decisions.

When considering an update, the CAB should assess the risks and impact of the update to the production environment and determine whether it should be deployed. In making this decision, it should consider the following issues.

- What else is happening in the operational environment?
- What is the impact of applying (versus not applying) the software update?
- What is the anticipated cost of deploying (versus not deploying) the software update?
- What are the steps that can be taken - if any - to mitigate the exposure to a security vulnerability or system instability, while the software update is being deployed?
- What is the impact of computer downtime? It may be necessary to compare and consider the risks of postponing the deployment of a software update versus the risks incurred by causing computer downtime when deploying a software update to your environment.
- What will be the best and most effective mechanism for deploying the software update?
- Are there any known issues or side effects with the software update, and does it necessitate a system restart?
- Are enough resources available to deploy the software update or deal with any issues experienced during deployment?
- How will you address any dependencies or prerequisites that need to be met before the software update can be deployed?

Although the best response to a software vulnerability is to deploy the software update that resolves the issue, sometimes it may be preferable to deploy a short-term countermeasure - such as closing network ports or shutting down external access to systems - while the software update is being rolled out to systems within the production environment. Applying such countermeasures may have several benefits.

- The majority of software updates require that target computers be restarted before the installation is complete and the computer is safeguarded. If you are prevented from immediately deploying a software update to your environment, because computer restarts are limited to specific maintenance windows, then implementing recommended countermeasures can effectively safeguard your computers until the software update can be deployed. Alternatively, sometimes security updates can be deployed and the automatic restart suppressed. In this instance, the security update could be installed during normal hours and then the computer restarted at a time suited to the maintenance window.

- Countermeasures may have a lower risk, and can be applied more quickly and with less testing than the software update itself. It may be significantly easier, for example, to disable network ports, or to shut down services or systems that are exposed to a particular security vulnerability, and apply the software update later.

Implementing computer hardening countermeasures can often protect computers from many common security vulnerabilities. Blocking certain network ports and disabling unused services are just two of the countermeasures that, when implemented, can effectively safeguard your computers. For more information on computer hardening countermeasures, see [6].

Even if countermeasures are deployed to reduce the exposure to a security vulnerability, the security update should still be scheduled for deployment. For example, if systems remain unpatched and a computer that is infected with a worm or virus is introduced into the network, the infection will quickly spread to all unprotected systems. The deployment of countermeasures should simply lower the priority of the change request, rather than remove the requirement for the software update.

3.3.1.4 Decide Who Will Own Deployment of the Software Update

Once an agreement has been reached to deploy the software update and to use any countermeasures (if appropriate), you need to identify who will take responsibility for making sure that these changes happen. This person will need to.

- Develop a plan for making the required changes.
- Determine and obtain the resources required.
- Arrange for the development of any necessary scripts, tools, and documentation that will be necessary to deploy the changes.
- Ensure that adequate testing is carried out.
- Ensure that the changes are deployed into production.
- Assess the success or failure of deployment.

Without a designated owner to oversee the above activities, there is a risk the software update might not be deployed.

3.3.2 Plan the Release

Release planning is the process of working out how to release the software update into the production environment.

There may be a number of issues and constraints that dictate the steps necessary to fully deploy the software update into production. For example, when tasked with deploying the software update, consider the following.

- How much time should users be given before the software update is installed automatically? The amount of time permitted will depend on a number of factors, including user roles and responsibilities, and the nature of the system instability or security vulnerability that the software update is designed to address.

- Certain software updates require administrative rights on the computer on which they are being installed. Most end users will not have local administrator rights, so the tool used to install the software update will need to be able to acquire elevated rights and privileges to install the software update on client computers.
- If the software update requires a certain amount of disk space to install, or the software update is to be cached locally prior to installation, then a check needs to be made on the amount of free disk space on each client computer.
- If the software update is large - for example, several megabytes in size – remote IT clients (such as those used by homeworkers) may take some time to download it. If the software update is not classified as an emergency, it may be appropriate to postpone installation on those clients until they are physically connected to the network.
- Business-critical computers may have specific times at which changes and computer restarts are permitted (outage windows). The deployment of a software update and any system restarts that are required as a result will need to be scheduled within these outage windows.
- Operating system security settings may need to be considered. For example, if Windows-based client computers are locked down through the use of Group Policy settings, this may affect the software update's ability to install correctly.
- If the product to be patched was deployed using separate installation software such as Microsoft’s Windows Installer, that software may seek access to the original installation files during the update. If the product was originally installed from physical media - a CD drive, for example – the installation software will try to find the original files on the currently inserted CD.

3.3.2.1 Writing a Release Plan

This is the point at which you will need to plan and determine the order in which the computers within your production environment will deploy the software update. The following are some of the issues you may need to consider when writing a release plan.

- If the software update applies to all servers within the production environment, which should be patched first? Patching the management infrastructure first ensures that administrators can use these services to monitor the progress of the deployment.
- Are there any business reasons why one part of the production environment should be patched before another? There may be compelling reasons to apply the software update to computers that are at risk from a security vulnerability or potential system instability and then, once these computers are patched, continue the rollout elsewhere.

- What impact does the available network bandwidth between sites have on the rollout order? It may be difficult to get the software update out to some sites as quickly as others, because of network bandwidth constraints. The software update can be deployed more quickly to sites with good network connectivity than to those where network availability is limited.

Finally, you will need to determine how and when information about the software update, its severity, impact, and the steps that need to be taken to deploy it, will be communicated to users, the business, and the service desk.

In the case where the change request is an emergency, then you should consider the following.

- If management architecture servers need to be patched, it may be appropriate to plan to have these computers patched manually - by local administrators - to ensure that these servers are not restarted during the rollout of the software update to other computers within the production environment.
- If one site or group of computers has already suffered from a security breach or system instability that the software update addresses, deployment of the software update should be directed to those computers first.

3.3.3 Build the Release

With a release plan written, the next stage of the process is to develop the scripts, tools, and procedures which the administrators will use to deploy the software update into the production environment.

If updates are already packaged in an executable format, there will be no need to perform any additional work to repackage them for deployment. Otherwise, it may be necessary to create a program to distribute and install the software update (tools or wizards supplied with the operating system may exist to automate this process).

3.3.4 Acceptance Testing

Up to this point, the purpose of testing has been to confirm that the software update and the release package work correctly within a development environment. The *acceptance testing* stage allows developers and business representatives to check that updates work in an environment that closely mirrors production and that business-critical systems will continue to run successfully once the software update has been deployed. Administrators, together with business representatives, should draw up a short set of tests that will be performed when a software update is regarded as business critical, and a more detailed set of tests that can be used when the software update has a lower priority.

However critical the software update is, a minimum level of testing should always be carried out to ensure that.

- Once installation is complete, the computer will reboot as it is designed to.

- The software update, if it is targeted at computers connected across slow or unreliable network connections, can be downloaded across these links and, once this completes, it successfully installs.
- The software update is supplied with an uninstall routine, which can be used to successfully remove the software update.
- Business or nationally critical systems and services continue to run once the software update has been installed.

Before you deploy the software update into the operational environment, it is important to collect information about any troubleshooting steps, procedures, and tools that are used during testing, and to make these available to service desk support staff and the operations team.
Ideally, that testing will result in the creation of.

- Documentation that describes standard troubleshooting steps, together with any associated workarounds.
- A list of contacts and an escalation path.
- Scripts, rules and information (such as counters, events, and thresholds), which will enable your operations staff to effectively monitor the release in production.

No matter how much testing is performed, rolling out a software update into production often produces effects that can never be anticipated or replicated in a lab environment. To avoid impacting a large number of client computers with a potential failure, administrators should consider rolling out the software update to a small representative sample of computers and confirming that business-critical systems and applications continue to run, before deploying it to the entire organisation.

3.3.5 Summary

The following are the key points to remember from the Evaluation, Planning and Testing phase.

- Use a formal process to determine whether it is in the best interests of the business to deploy the software update.
- Identify an owner of the software update who will be responsible for ensuring that it is deployed.
- When the deployment of the software update is approved, it is necessary to plan how to deploy it across the organisation.
- Test the update in a reference system and, if needed, pilot test it in a production environment before deployment to confirm that it does not compromise critical business applications.

3.4 The Deployment Phase

The Deployment phase focuses on the tasks and activities required to deploy a software update into the operational environment. The deployment of a software update consists of the following activities.

- Deployment preparation.
- Deployment of the software update to targeted computers.
- Post-implementation review.

3.4.1 Deployment Preparation

The production environment needs to be prepared for each new release. The most important aspect of this is informing end users and administrators about the impending release of an update. Ideally, send a clear and easily identifiable e-mail message to users and administrators, which both notifies them of the update and provides information about how to install it. Flag the mail for follow-up before sending it to remind users and administrators of the actions they need to take.

If you are deploying an update to desktops outside of core business hours, the e-mail message should tell users to leave their computers on overnight on a specified date (if this is not already standard practice).

3.4.2 Deployment of the Software Update to Targeted Computers

The process for deploying the software update into the operational environment will depend on the type and nature of the release, as well as the release mechanism selected.
It will also depend significantly on whether the software update is an emergency. Because of the urgency associated with emergency changes, there will be some differences in how you deploy them. Those differences will be highlighted throughout this section.

Ideally, you should release software updates through a phased deployment, which minimises the impact of any failures or adverse effects that might be introduced by the initial distribution of a software update.

Computers can fail to install an update for several reasons, such as.

- The computer is offline.
- The computer is being rebuilt or re-imaged.
- The computer has insufficient disk space.

If you face exceptions during a normal deployment, you should have sufficient time to stop, determine the root cause, and re-deploy. However, during an emergency deployment, the time available for triage and root cause assessment will be much shorter. In both cases, it is important to have a plan in place to stop the rollout, uninstall failed updates, and then redeploy those updates.

3.4.3 Post-Implementation Review

The post-implementation review should typically be conducted within one to four weeks of a release deployment to identify improvements that should be made to the patch management process. The review should.

- Ensure that the vulnerabilities identified by the original vulnerability notification are added to your vulnerability scanning reports and security policy standards, so the attack does not have an opportunity to recur.
- Ensure that your build images have been updated to include the latest software updates following the deployment.
- Discuss planned versus actual results.
- Discuss the risks associated with the release.
- Review your organisation's performance throughout the incident. Take this opportunity to improve your response plan to include any lessons learned.
- Discuss changes to your service windows.
- Assess the total incident damage and costs, including both downtime and recovery costs.
- Create another baseline or update the existing baseline for your environment.

3.4.4 Summary

During the deployment phase, you should have accomplished the following key activities.

- Established the order in which the software updates will be rolled out into production.
- Surveyed your production environment to ensure it would be able to handle the software update.
- Placed the software update files on SMS distribution points or SUS servers (for Windows environments).
- Deployed the software update into production.
- Rescanned your environment to assess success, and patched any computers that failed to install the software update.
- Performed a review of the patch management process once the deployment is complete.

Deployment of the software update is not the end of the process. The assessment and inventory phase is continuous, and should be returned to following a deployment (if only to update the IT inventory accordingly).

4 Patch Management Metrics[2]

There are three main categories of patch and vulnerability metrics: susceptibility to attack, mitigation response time, and cost. This section provides example metrics in each category.

4.1 Measuring a System's Susceptibility to Attack

An organisation's susceptibility to attack can be approximated by several measurements. An organisation can measure the number of patches needed, the number of vulnerabilities, and the number of network services running on a per system basis. These measurements should be taken individually for each computer within the system, and the results then aggregated to determine the system-wide result.

Both raw results and ratios (e.g. number of vulnerabilities per computer) are important. The raw results help reveal the overall risk a system faces because the more vulnerabilities, unapplied patches, and exposed network services that exist, the greater the chance that the system will be penetrated. Large systems consisting of many computers are thus inherently less secure than smaller similarly configured systems. This does not mean that the large systems are necessarily secured with less rigor than the smaller systems. To avoid such implications, ratios should be used when comparing the effectiveness of the security programs of multiple systems. Ratios (e.g., number of unapplied patches per computer) allow effective comparison between systems. Both raw results and ratios should be measured and published for each system, as appropriate, since they are both useful and serve different purposes.

The initial measurement approach should not take into account system security perimeter architectures (e.g. firewalls) that would prevent an attacker from directly accessing vulnerabilities on system computers. This is because the default position should be to secure all computers within a system even if the system is protected by a strong security perimeter. Doing so will help prevent insider attacks and help prevent successful external attackers from spreading their influence to all computers within a system.

Recognising that most systems will not be fully secured, for a variety of reasons, the measurement should then be recalculated while factoring in a system's security perimeter architecture. This will give a meaningful measurement of a system's actual susceptibility to external attackers. For example, this second measurement would not count vulnerabilities, network services, or needed patches on a computer if they could not be exploited through the system's main firewall.

While the initial measurement of a system's susceptibility to attack should not take into account the system security perimeter architecture, it may be desirable to take into account an individual computer's security architecture. For example, vulnerabilities exploitable by network connections might not be counted if a computer's personal firewall would prevent such exploit attempts. This should be

[2] This section is largely drawn from the NIST white paper on Patch Management [1], which may be consulted for further information.

done cautiously because a change in a computer's security architecture could expose vulnerabilities to exploitation.

4.1.1 Number of Patches

Measuring the number of patches needed per system is natural for organisations that have deployed enterprise patch management tools, since these tools automatically provide such data. The number of patches needed is of some value in approximating an organisation’s susceptibility to attack, but its effectiveness is limited because a particular security patch may fix one or many vulnerabilities, and these vulnerabilities may be of varying levels of severity. In addition, there are often vulnerabilities published for which there are no patches. Such vulnerabilities intensify the risk to organisations, yet are not captured by measuring the number of patches needed. The quality of this measurement can be improved by factoring in the number of patches rated critical by the issuing vendor and comparing the number of critical and non-critical patches.

4.1.2 Number of Vulnerabilities

Measuring the number of vulnerabilities that exist per system is a better measure of an organisation's susceptibility to attack, but still is far from perfect. Organisations that employ vulnerability scanning tools are most likely to employ this metric, since such tools usually output the needed statistics. As with measuring patches, organisations should take into account the severity ratings of the vulnerabilities, and the measurement should output the number of vulnerabilities at each severity level (or range of severity levels). Vulnerability databases, vulnerability scanning tools, and the patch vendors themselves usually provide rating systems for vulnerabilities; however, currently there is no standardised rating system. Such rating systems only approximate the impact of a vulnerability on a stereotypical generic organisation. The true impact of a vulnerability can only be determined by looking at each vulnerability in the context of an organisation's unique security infrastructure and architecture. In addition, the impact of a vulnerability on a system depends on the network location of the system (i.e., when the system is accessible from the Internet, vulnerabilities are usually more serious).

4.1.3 Number of Network Services

The last example of an attack susceptibility metric is measuring the number of network services running per system[3]. The concept behind this metric is that each network service represents a potential set of vulnerabilities, and thus there is an enhanced security risk when systems run additional network services. When taken on a large system, the measurement can indicate a system’s susceptibility to network attacks (both current and future). It is also useful to compare the number of network services running between multiple systems to identify systems that are doing a better job at minimising their network services. Having a large number of network services active is not necessarily indicative of system administrator mismanagement.

[3] Organisations should consider assigning weights to services or network ports when counting them, because they may not all be equally important. For example, a single network port could be used by multiple services. Also, one service might be much more likely to be attacked than another or might perform much more important functions than another.

However, such results should be scrutinised carefully to make sure that all unneeded network services have been turned off.

4.2 Mitigation Response Time

It is also important to measure how quickly an organisation can identify, classify, and respond to a new vulnerability and mitigate the potential impact within the organisation. Response time has become increasingly important, because the average time between a vulnerability announcement and an exploit being released has decreased dramatically in the last few years. There are three primary response time measurements that can be taken: vulnerability and patch identification, patch application, and emergency security configuration changes.

4.2.1 Response Time for Vulnerability and Patch Identification

This metric measures how long it takes the responsible officer (e.g. system administrator, IT operations manager) to learn about a new vulnerability or patch. Timing should begin from the moment the vulnerability or patch is publicly announced. This measurement should be taken on a sampling of different patches and vulnerabilities and should include all of the different resources the responsible officer uses to gather information.

4.2.2 Response Time for Patch Application

This metric measures how long it takes to apply a patch to all relevant IT devices within the system. Timing should begin from the moment the responsible officer becomes aware of a patch. This measurement should be taken on patches where it is relatively easy for the responsible officer to verify patch installation. This measurement should include the individual and aggregate time spent for the following activities.

- Patch analysis
- Patch testing
- Configuration management process
- Patch deployment effort.

Verification can be done through the use of enterprise patch management tools or through vulnerability scanning (both host and network-based). It may be useful to take this measurement on both critical and non-critical security patches, since a different process is usually used by organisations in both cases, and the timing will likely be different.

4.2.3 Response Time for Emergency Configuration Changes

This metric applies in situations where a vulnerability exists that must be mitigated but where there is no patch. In such cases the organisation is forced to make emergency configuration changes that may reduce functionality to protect the organisation from exploitation of the vulnerability. Such changes are often done at the firewall, e-mail server, Web server, central file server, or servers in the DMZ. The changes may include turning off or filtering certain e-mail attachments, e-mail subjects, network ports, and server applications. The metric should measure the time it takes from the moment the responsible officer learns about the vulnerability to the moment that an acceptable workaround has been applied and verified. Because many vulnerabilities will not warrant emergency configuration changes, this metric will be for a subset of the total number of vulnerabilities for any system.

These activities are normally done on an emergency basis, so obtaining a reasonable measurement sample size may be difficult. However, given the importance of these activities, these emergency processes should be tested, and the timing metric can be taken on these test cases. The following list contains examples of emergency processes that can be timed.

- Firewall or router configuration change
- Network disconnection
- Intrusion prevention device activation or reconfiguration
- E-mail filtering rules addition
- Computer isolation
- Emergency notification of staff.

The metric results are likely to vary widely between systems, since the emergency processes being tested may be very different. As much as possible, organisations should create standard system emergency processes, which will help make the testing results more uniform. Organisations should capture and review the metrics following any emergency configuration change as a part of an operational debriefing to determine subsequent actions and areas for improvement in the emergency change process.

4.3 Cost

Measuring the cost of patch and vulnerability management is difficult because the actions are often split between many different personnel and groups. Most organisations will have the patch and vulnerability functions split between multiple groups and allocated to a variety of full-time and part-time personnel. There are three main cost measurements that should be taken: system administration support, enterprise patch and vulnerability management tools, and incidents that occurred due to failures in the patch and vulnerability management program.

4.3.1 Cost of System Administration

This measurement is always difficult to take with accuracy but is important nonetheless. The main problem is that, historically, system administrators have not been asked to calculate the amount of time they spend on security, much less on security patch and vulnerability management. As organisations improve in their overall efforts to measure the real cost of IT security, measuring the cost of patch and vulnerability measurement with respect to system administrator time will become easier.

4.3.2 Cost of Enterprise Patch and Vulnerability Management Tools

This measurement includes patching tools, vulnerability scanning tools, vulnerability Web portals, vulnerability databases, and log analysis tools (used for verifying patches). It should not include intrusion detection, intrusion prevention, and log analysis tools (used for intrusion detection). Organisations should first calculate the purchase price and annual maintenance cost for each software package. Organisations should then calculate an estimated annual cost that includes software purchases and annual maintenance. To create this metric, the organisation should add the annual maintenance cost to the purchase price of each software package divided by the life expectancy (in years) of that software. If the software will be regularly upgraded, the upgrade price should be used instead of the purchase price.

As a rule, the estimated annual cost equals the sum of annual maintenance for each product plus the sum of the amortised purchase price for each product. The amortised purchase price is defined as the purchase price (or upgrade price) of the product divided by the product's estimated operational life ("life expectancy") in years.

For example, an organisation has the following software.

Product	Purchase price	Upgrade price	Life expectancy	Annual maintenance
Enterprise patch management software	£30K	£15K	4 years	£3K
Vulnerability scanner	£20K	£10	3 years	£2K

Assume that the organisation plans to upgrade the vulnerability scanner software after three years, but plans to switch to new enterprise patch management software after four years. The estimated annual cost will be (£3K + £2K) + (£30K/4) + (£10K/3) = £15,833.

4.3.3 Cost of Program Failures

This measurement calculates the total cost of the business impact of all incidents that could have been prevented if the patch and vulnerability mitigation program had been more effective, as well as all problems caused by the patching process itself, such as a patch inadvertently breaking an application. The cost numbers should include tangible losses (, worker time and destroyed data) as well as intangibles (e.g. placing a value on an organisation's reputation). It should be calculated on an annual basis. The results of this measurement should be used to help evaluate the cost effectiveness of the patch and vulnerability management program. If the cost of program failures is extremely high, then the organisation may be able to save money by investing more resources in their patch and vulnerability management program. If the cost of program failures is extremely low, then the organisation can maintain the existing level of support for patch and vulnerability management or possibly even decrease it to optimise cost effectiveness.

4.3.4 Performance Targets and Cost Effectiveness

Realistic performance targets for each metric should be communicated to system owners and system security officers. Once these targets have been achieved, more ambitious targets can be set. It is important to carefully raise the bar on patch and vulnerability security to avoid overwhelming system security officers and system administrators.

The cost effectiveness of a program can be calculated by comparing the cost metrics associated with running the program to the cost of program failures. It can also be calculated by comparing the cost metrics associated with running the program to the metrics that indicate program performance (the response time and susceptibility to attack metrics).

4.4 Summary

Every organisation should consistently measure the effectiveness of its patch and vulnerability management program and apply corrective actions as necessary. This can be done by developing a patch and vulnerability metrics program. The metrics should be targeted toward the patch and vulnerability management program's maturity level, with particular metrics being most valuable for certain maturity levels. Organisations should document which metrics will be taken for each system and should document the details of each of the metrics. Realistic performance targets should be communicated to system owners and system security officers.

5 Sources of Further Advice

5.1 WARPs

WARPs (Warning, Advice and Reporting Points) are part of NISCC's information sharing strategy to protect the UK's Critical National Infrastructure from electronic attack. WARPs have been shown to be effective in improving information security by stimulating better communication of alerts and warnings, improving awareness and education, and encouraging incident reporting.

For further information on WARPs, visit the WARP homepage at http://www.warp.gov.uk/home.htm

Patch management is a common topic for WARPs to cover. Existing WARPs have.

- Used WARP bulletin boards and e-mail lists to share experiences of patch management within their community.
- Collaborated to filter software update notifications and advisory bulletins.

5.2 Uniras

Uniras, as the UK Government's Computer Emergency Response Team (CERT), provides government and CNI organisations with support in responding to electronic attack incidents. This may vary from answering queries via the telephone to onsite assistance. It is administered by NISCC.

Team members are specialists in the field of IT Security Incident Management and besides their normal day-to-day activities regularly lecture in incident management on courses for the NISCC community. They also provide assistance to other organisations wishing to set up Incident Response Teams.

Uniras can be contacted by e-mail at uniras@niscc.gov.uk. Further information about Uniras is available at http://www.uniras.gov.uk/niscc/reportIncident-en.html

6 Appendix A: Native Windows Patch Management Tools[4]

When Microsoft is made aware of a security vulnerability, the issue is evaluated and verified by the Microsoft Security Response Centre (MSRC) and the appropriate product groups. The MRSC then creates and tests a security patch to remedy the issue, and works with the reporter of the vulnerability to coordinate the release of public information in the form of a security bulletin that has the security patch details.

Microsoft then distributes the software update through the Microsoft Download Centre and other services, including.

- Microsoft Windows Update
- Microsoft Office Update
- Microsoft Software Update Services (SUS)
- Microsoft Server Update Services (WSUS, not to be confused with SUS)
- Microsoft Systems Management Server (SMS) 2.0 with the SUS Feature Pack
- Microsoft Systems Management Server (SMS) 2003

As the software update is about to be released, the MSRC sends out a related security bulletin.

Typically, security patches are made available for supported products not only on the current service pack, but also the one previous. However, this is not always the case, so you should check the product support life cycle policies for your products to be sure.

6.1.1 Software Update Terminology

Table 2 below lists Microsoft terminology for software updates. Note that Microsoft no longer use the term “patch”, except as part of the term “security patch” or when describing the process of patch management.

[4] This section is derived from information available on the Microsoft website: see [1].

Term	Definition
Security patch	A broadly released fix for a specific product, addressing a security vulnerability. A security patch is often described as having a severity, which actually refers to the MSRC severity rating of the vulnerability that the security patch addresses.
Critical update	A broadly released fix for a specific problem, addressing a critical, non-security related bug.
Update	A broadly released fix for a specific problem, addressing a non-critical, non-security related bug.
Hotfix	A single package composed of one or more files used to address a problem in a product. Hotfixes address a specific customer situation, are only available through a support relationship with Microsoft, and may not be distributed outside the customer organisation without written legal consent from Microsoft. The terms QFE (Quick Fix Engineering update), patch, and update have been used in the past as synonyms for hotfix.
Update rollup	A collection of security patches, critical updates, updates, and hotfixes, which are released as a cumulative offering or targeted at a single product component, such as Microsoft Internet Information Services (IIS) or Microsoft Internet Explorer. Allows for easier deployment of multiple software updates.
Service pack	A cumulative set of hotfixes, security patches, critical updates, and updates since the release of the product, including many resolved problems that have not been made available through any other software updates. Service packs may also contain a limited number of customer-requested design changes or features. Service packs are broadly distributed and tested by Microsoft more than any other software updates.
Integrated service pack	The combination of a product with a service pack in one package.
Feature pack	A new feature release for a product that adds functionality. Usually rolled into the product at the next release.

Table 2 - Microsoft Terminology for Software Updates

6.1.2 Tools and Technologies

This section will examine the automated tools that organisations of all sizes can use to manage and control software update installation. There are three principal Microsoft technologies available for enterprise patch management of Windows-based systems.

- Software Update Services (SUS)

- Windows Server Update Services (WSUS), a recent extension of SUS.
- Systems Management Server (SMS)

6.1.3 Software Update Services (SUS)

SUS is a free tool that allows you to install a service to download all critical updates, security updates, and service packs as they are posted to the Microsoft Windows Update web site at http://windowsupdate.microsoft.com/en/default.asp. As you approve each update, SUS will automatically make them available to all preconfigured servers running Microsoft Windows Server 2003 and Windows 2000, as well as to desktops running Windows XP Professional and Windows 2000 Professional. SUS supports only critical and security updates - including service packs - that apply to the operating system and components included with the operating system. All other software updates - including those for applications- need to be handled using a different mechanism.

SUS includes the following capabilities.

- Software updates can be approved uniquely on each SUS server; enabling testing in a separate environment, as well as phased deployments across the enterprise.
- Software updates can be distributed through SUS (saving bandwidth on shared Internet connections), or SUS clients can be configured to download software updates from the Windows Update site.
- SUS can provide Windows Update software updates to computers that do not have Internet access.
- SUS can scale to very large environments, because the SUS server architecture is made up of simple parent/child relationships and each SUS server can support up to 15,000 clients.
- Software updates can be copied by CD from an SUS server that is connected to the Internet, to an SUS server architecture with no Internet access.

SUS servers require the Microsoft Windows Server 2003 operating system or Windows 2000 Server, Internet Information Services, and port 80 for communications with SUS clients. Every SUS server can be configured to synchronise software update packages and approvals, either manually or automatically from its parent SUS server, enabling flexibility in how the environment is maintained.

SUS clients use the Automatic Updates client (which is also used by Windows Update). Clients are configured to connect to specific servers, and can be configured for automatic software update installations or end-user prompting.

To determine whether any computers have failed to install the updates that have been made available to them through SUS, you should run the Microsoft Baseline Security Analyser (MBSA) on a periodic basis. MBSA scans for missing security updates and reports on a computer's adherence to common security best practices (such as strong passwords), and identifies any configuration options that leave the computer open to potential security vulnerabilities. MBSA can also be configured to report on updates that have already been approved on an SUS server, but have not yet been installed.

6.1.4 Windows Server Update Services (WSUS)

WSUS is a recent product offering from Microsoft that claims to offer additional features beyond those provided by SUS, including.

- A larger menu of updates;
- The ability to automatically download updates from Microsoft Update by product and type;
- Additional language support for customers worldwide;
- Maximized bandwidth efficiency through Background Intelligent Transfer Service (BITS) 2.0;
- The ability to target updates to specific computers and computer groups;
- The ability to verify that updates are suitable for each computer before installation;

  More flexible deployment options, reporting capabilities and database options;
- Data migration and import/export capabilities; and
- Extensibility through the application programming interface (API).

Microsoft plans WSUS to ultimately replace SUS. However, at the time of writing the company has no plans to remove support for SUS.

6.1.5 Systems Management Server (SMS)

Microsoft Systems Management Server (SMS) is designed to deploy and manage the distribution of software updates to a large number of clients. It provides the following functionality.

- Inventory functions to determine how many computers have been deployed and to identify their locations and roles.
- Inventory functions to identify which software applications and software updates have been installed and which need to be installed on the deployed computers.
- Scheduling functions that allow an organisation to deploy software updates outside regular working hours, or at a time that has the least impact on business operations.
- Status reporting that allows administrators to monitor installation progress.

The SMS 2003 inventory scanning programs are used to create an inventory of applicable and installed updates for each client computer, using an automated source of detection logic. The resulting data is included in the Systems Management

Server inventory and a comprehensive view of the status is provided through the Web-based reporting capabilities. Typically, the inventory data will be limited to those items that are released by Microsoft as security bulletins.

More information on SMS can be found at http://www.microsoft.com/smserver.

7 Appendix B: Patch Management Tools for Other Operating Systems

7.1 Linux

System administrators responsible for patch management on Linux systems should note that the applications and utilities used for patch management vary with the Linux distribution.

Red Hat Enterprise Linux, Fedora, SUSE and Debian distributions all have different patch management tools available, including up2date (Red Hat), *apt* (Debian), *yum* (Red Hat*)*, YaST online update (SUSE) and Zenworks Linux Management (Novell). Administrators may therefore wish to become familiar with the patch management tools available for the distribution(s) they are responsible for.

Michael Jang's book "Linux Patch Management: Keeping Linux Systems Up To Date" [5] is a good starting point for all Linux administrators who wish to become more familiar with Linux patch management techniques and tools.

7.1.1.1 Solaris

Solaris Patch Manager is Sun's unified patch management offering for Solaris. It is available in two versions, both based on the same patch analysis engine.

- **Solaris Patch Manager Base Version 1.0 for the Solaris 2.6, 7 and 8 Operating Environments** is available free of charge through the SunSolve Online web site. Base Version 1.0 comes with a command line interface (CLI) and performs the following tasks.
    - Determination of required patches for a system
    - Automatic patch download
    - Automatic simple patch installation
    - Resolution of patch dependencies
    - Specification of install order
    - Removal of patches
- **Solaris Patch Manager 1.0** comes bundled with the Solaris 9 Operating System. In addition to the functions listed above for Base Version 1.0, Solaris Patch Manager 1.0 comes with a graphical user interface (GUI) and the ability to.

- Perform remote patch management on other systems with the Solaris 9 Operating Environment
- Automatically install a patch or list of patches to homogeneous systems running the Solaris 9 Operating Environment

Initially, only patches for the Solaris Operating Environment, Network Storage products, Sun Cluster software, Sun Enterprise 10000 and 15000 servers, and Sun Fire servers will be provided by Solaris Patch Manager. Over time, patches for additional Sun products will also be supported.

8 Appendix C: Patching Network Infrastructure

As network infrastructure components such as firewalls and routers become more ever more complex devices, there is an increasing requirement to patch these devices as well as the actual computers on the network. Patching network infrastructure presents specific challenges, notably the relative immaturity of centralised vulnerability reporting and automated patch management tools when compared to the server environment.

Patches for infrastructure components are typically made available by vendors in the same way as operating system patches. The two companies who dominate the network infrastructure market, Cisco and Juniper, make software upgrades available on the following sites respectively.

- The Cisco Software Download Centre, http://www.cisco.com/kobayashi/sw-center/
- Juniper’s Support Centre, http://www.juniper.net/support

Note that access to both sites is restricted to registered customers of these companies.

Additionally, network infrastructure vendors may provide ancillary tools to assist in patch management. For example, Cisco’s IOS Upgrade Planner is designed to assist users in selecting the correct software updates for their individual infrastructure components.

Traditionally, updating infrastructure components has required the network administrator to manually send the updates over the network to the relevant devices by means of the Simple Network Management Protocol (SNMP), Secure Shell (SSH) or Telnet. However, infrastructure vendors are increasingly moving towards the more automatic style of patch management familiar from the server (especially Windows server) environment. System administrators using SNMP, SSH or Telnet to perform manual updates should familiarise themselves with known vulnerabilities in these protocols, as listed in Uniras and vendor advisory notices.

9 References

[1] "Creating a Patch and Vulnerability Management Program", NIST Special Publication 800-40 (version 2.0), November 2005. See http://csrc.nist.gov/publications/nistpubs/

[2] Microsoft's Security Guidance for Patch Management. http://www.microsoft.com/technet/security/topics/patchmanagement.mspx

[3] HMG Infosec Standard No. 2, "Risk Management and Accreditation of Information Systems", Issue 2.0, July 2005. Also available as a NISCC document at http://www.niscc.gov.uk/niscc/docs/re-20050804-00653.pdf

[4] "Essentials of Patch Management Policy and Practice", Jason Chang, Jan 2004. See http://www.patchmanagement.org/pmessentials.asp

[5] Michael Jang, "Linux Patch Management: Keeping Linux Systems Up To Date", January 2006. ISBN 0132366754.

[6] The Microsoft Threats and Countermeasures Guide. http://www.microsoft.com/downloads/details.aspx?displaylang=en&familyid=1b6acf93-147a-4481-9346-f93a4081eea8

[7] NISCC First Responders Guide: Policy and Principles, v1.2, October 2005. http://www.niscc.gov.uk/niscc/docs/re-20051004-00868.pdf.

[8] NISCC Technical Note 08/04, "Introduction to Vulnerability Assessment Tools", October 2004. http://www.niscc.gov.uk/niscc/docs/re-20041006-00750.pdf