CMSGu2014-03

National Computer Board

Mauritian Computer Emergency Response Team

Enhancing Cyber Security in Mauritius

Guideline on Vulnerability and Patch Management

CERT-MU

National Computer Board
Mauritius

September 2014

Issue No. 3

Version 2.0

Table of Contents

DISCLAIMER: *This guideline is provided "as is" for informational purposes only. Information in this guideline, including references, is subject to change without notice. The products mentioned herein are the trademarks of their respective owners.*

1.0 Introduction

1.1 Purpose and Scope

The purpose of this guideline is to assist organisations in identifying the vulnerabilities in their IT systems and patch them accordingly. It focuses on how to create an organisational model and test its effectiveness. It also covers technical solutions that are available for vulnerability management.

1.2 Audience

The target audience for this document include security managers responsible for designing and implementing security patch and vulnerability remediation strategies. System administrators and security operations officers who are responsible for applying patches and deploying solutions are also targeted.

1.3 Document Structure

This document is organised into the following sections:

Section 1 outlines the document's content, the targeted audience and the document's structure.

Section 2 gives a background on vulnerability and patch management.

Section 3 elaborates on the importance of a vulnerability and patch management process.

Section 4 concludes the document.

Section 5 comprises a list of references that have been used in this document.

Appendix A defines a set of acronyms used in this document.

2.0 Background

A vulnerability in a system is a flaw that can be exploited by a malicious user so as to gain unauthorised access to the system. A patch is an additional piece of code written to remove 'bugs' in system software. A patch normally addresses security flaws within a program.

Not all vulnerabilities that exist have correlated patches. Thus, system administrators must not only be aware of applicable vulnerabilities and available patches, but also other methods of remediation (for example device or network configuration changes, staff training and education) that reduce the exposure of systems to vulnerabilities.

Vulnerability and patch management is a method adopted by security professionals to proactively prevent the exploitation of IT vulnerabilities that exist within an organisation. This is primarily done to minimise the time and cost of dealing with vulnerabilities and exploitation of those vulnerabilities. Proactively managing vulnerabilities found in systems will trim down or eradicate the potential for future system exploitation and involve significantly less time and effort than responding to the incident after the damage has been caused.

Timely patching of security issues is a critical process to maintain the operational availability, confidentiality, and integrity of IT systems. However, failure to keep operating system and application software patched is one of the most common issues identified by security and IT professionals. New patches are released daily, and it is often difficult for even experienced system administrators to keep abreast of all the new patches and ensure proper deployment in a timely manner.

Most major attacks in the past few years have targeted known vulnerabilities for which patches existed before the outbreaks. Indeed, the moment a patch is released, attackers make a rigorous effort to reverse engineer the patch within days or even hours, identify the vulnerability, and develop and release exploit code. Thus, the time immediately after the release of a patch is a vulnerable moment for most organisations due to the time lag in obtaining, testing, and deploying a patch.

To help address this escalating issue, it is recommended that all organisations have a systematic, accountable, and documented process for managing exposure to vulnerabilities through the timely deployment of patches.

3.0 The Importance of a Patch and Vulnerability Management

3.1 Why Patch and Vulnerability Management is required?

The rise in cybercrime and the associated risks are compelling most organisations to focus on information security. A patch and vulnerability management process should be part of an organisation's effort to control information security risks. This process will allow an organisation to obtain a continuous overview of vulnerabilities in their IT environment and the risks associated with them. It is only by identifying and mitigating vulnerabilities in the IT environment that an organisation can prevent attackers from penetrating their networks and stealing information.

3.2 Developing a Patch and Vulnerability Management Process

3.2.1 Objective

The main objective of a patch and vulnerability management process is to detect vulnerabilities and patch them in a timely manner. Many organisations do not frequently perform vulnerability scans in their environment. They perform scans on a quarterly or annual basis which only provides a snapshot at that point in time. The figure below shows a possible vulnerability lifecycle with annual scanning in place:

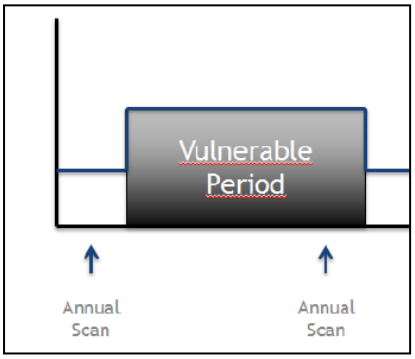

Figure 1 Vulnerability Scanning

Any vulnerability not detected after a schedule scan takes place, will only be detected at the next scheduled scan. This could leave systems vulnerable for a long period of time. When implementing a vulnerability management process, regular scans should be scheduled to reduce the exposure time. The above situation will then look like this:

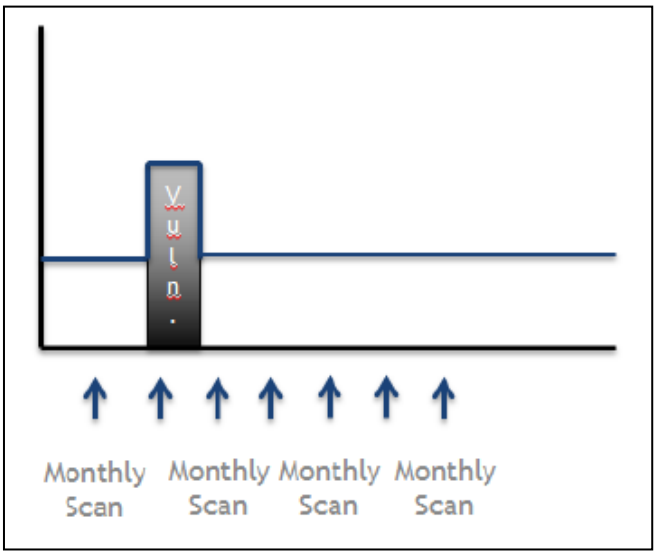

Figure 2 Continuous Vulnerability Management

Regular scanning ensures new vulnerabilities are detected in a timely manner; allow them to be patched faster. Having this process in place significantly reduces the risks of an organisation.

3.2.2 Vulnerability Scanners

It is not only important to understand how vulnerability scans are performed, but also what tools are available on the market. Today, little technical expertise is required to operate a vulnerability scanning tool. The majority of vulnerability scanners can be controlled through a Graphical User Interface (GUI) that allows a user to launch vulnerability scans against an entire network with a few mouse clicks. Several vendors provide various technical solutions, with different deployment options. These deployment options include standalone, managed services or even software as a service (SaaS). Some of the vendors offering vulnerability scanning technology include: McAfee, Qualys, Rapid 7, Tenable Network Security.

It is recommended that an organisation carefully tests vulnerability scanning products before deciding which solution best meets their requirements. Proper consideration should be given to the fact that scanning a single box with multiple products using their default settings could produce very different results. No matter which vulnerability scanning solution is selected, it is important to properly configure and customise scans to limit the amount of false positives in the scan results.

3.2.3 Associated risks

Since vulnerability scanning typically involves sending a large number of packets to systems, they might sometimes trigger unusual effects such as, for example, disrupting network equipment, causing downtime. However, since vulnerability scanning is mainly limited to scanning and not exploiting, risks are minimal. In order to cover these risks, it is always important to inform various stakeholders within an organisation when vulnerability scanning is taking place.

3.3 Roles and responsibilities

When building a patch and vulnerability management process, the following roles should be identified within the organisation:

1. **Security Officer:** The security officer is the owner of the vulnerability and patch management process. This person designs the process and ensures it is implemented as designed.
2. **Vulnerability Engineer (or System Administrator):** The vulnerability engineer is responsible for configuring the vulnerability scanner and scheduling the various vulnerability scans.
3. **Asset Owner:** The asset owner is responsible for the IT asset that is scanned by the vulnerability management process. This role should decide whether identified vulnerabilities are mitigated or their associated risks are accepted.
4. **IT System Engineer:** The IT system engineer is typically responsible for implementing corrective actions defined as a result of detected vulnerabilities.

3.4 Vulnerability Analysis

Vulnerability analysis, in relation to patch management, is the process of determining when and if a patch should be applied to a system. It is recommended that a security team (comprising of the roles mentioned above) be used to analyze and determine whether or not the system is vulnerable to identified attacks. A method used to determine if a system is vulnerable to an identified attack is through the use of the “vulnerability footprint,” also known as the attack surface.

The vulnerability footprint consists of four individual key elements (Impact, Exposure, Deployment, and Simplicity) that create a graphical representation of the vulnerability

footprint in the shape of a diamond (see Figure 3). The larger the physical size of the footprint, the more vulnerable the system is to attacks and the more urgent it becomes to mitigate that vulnerability. The relative shape of the diamond gives a graphical representation of main risk factors, where larger parts of the diamond correspond to a greater impact to risk.

The following elements define the vulnerability footprint and can be used by asset owners in determining the vulnerability of their specific system configurations:

- **Deployment**

  This element rating gives the relative proportion of systems installations having critical infrastructure and key resources that may contain vulnerable configurations at one site. A high rating would indicate all, or at least a high number, of deployed system at the asset owner's site are affected. A low rating indicates that only a few minor systems are exposed. This rating is important in that the answer provides an immediate yes or no determination of whether patching should be done and indicates whether this vulnerability affect the asset owner's system or not.

- **Exposure**

  This element rating ranks available layers of defense such as defense-in-depth and existing adequate barriers (the exploit affects the asset owner's system and is readily available to attackers). A high exposure rating indicates that an attacker can gain unauthenticated access to the system from another less-secure network within the control systems perimeter. A medium rating indicates that an attacker can gain unauthenticated remote access. A low rating indicates that an attacker can only gain authenticated physical machine/network access. Exposure of the system to unauthorized access presents significant risk.

- **Impact**

  A high impact element rating indicates that an exploit is successfully deployed into the wild and an attacker can gain full system control. A medium rating indicates that an attacker can obtain limited access or gain enough information to launch a denial-of-service (DoS) attack. A low rating indicates that an attacker gains enough information for a preliminary reconnaissance effort on a target system's architecture. Part of the impact assessment must consider cascade effects on safety and protection

devices. The initial penetration may not be immediately significant, but safety and production components could be disabled in the same system due to cascade effects from exhaustion of computer resources.

- **Simplicity**

  This rating applies to relative ease of the technical exploit. A high rating indicates an exploit that is written, available, and only requires average or basic computer skills to use (for example, an online script is available to implement the exploit). A medium rating indicates that a vulnerability exists, but original work needs to be done to use the exploit. A low rating indicates that the exploit requires a high level of computer skill and related knowledge.

Figure 3 shows Medium Deployment, High Exposure, Medium Impact, and Low Simplicity. The highest vulnerability is Exposure from unauthenticated outside attacks.

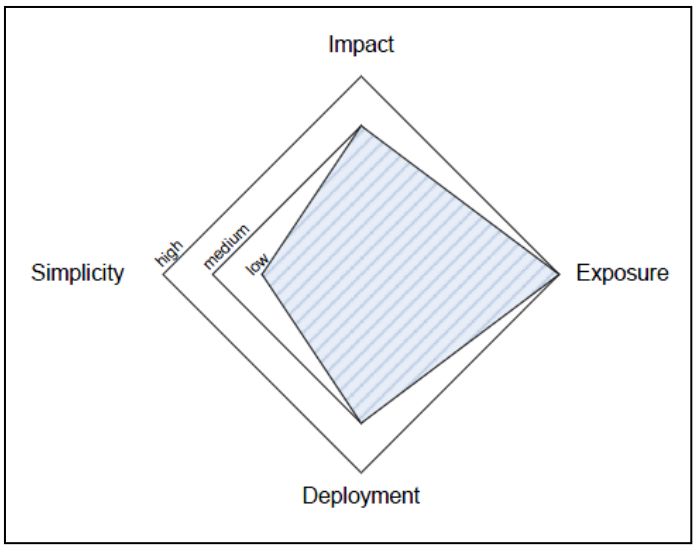

Figure 3 Example of a vulnerability footprint

This example is a heap-based protocol server overflow that allows remote attackers to cause a DoS attack (crash) and possibly execute arbitrary code via a crafted request for a file that causes a 'strncpy' call. In this example, Deployment is Medium (other systems are vulnerable to attack from the first system), the Exposure is High (vulnerability from remote, unauthenticated attackers), the Impact is Medium (attacker can cause a DoS attack), and Simplicity is Low (details of the exploit had not yet been published).

3.5 The Patch Process

The flowchart in Figure 4 shows the basic decision process in determining the urgency to patch the system. A documented process should be in place to monitor new exploits and vulnerabilities. When a vulnerability and patch has been identified, the asset owner should determine if it affects any system in the operation. If it does affect one or more systems, then a workaround should be considered. If a workaround is found, then the patch should be evaluated and scheduled as part of the regular patch cycle. If there are no workarounds, then the security team will have to analyze the risk associated with the patch. Factors that are considered in the analysis include the key elements of the vulnerability footprint measured against the potential impact to the business operations. If the risk is high, then an immediate patch may be required. Alternatively, if there are strong business constraints or operational concerns related to implementing the patch at a specific time, then it may be necessary to hold off on patching the system until the scheduled maintenance window. Once the patch has been implemented all applicable documentation and patch records should be updated.

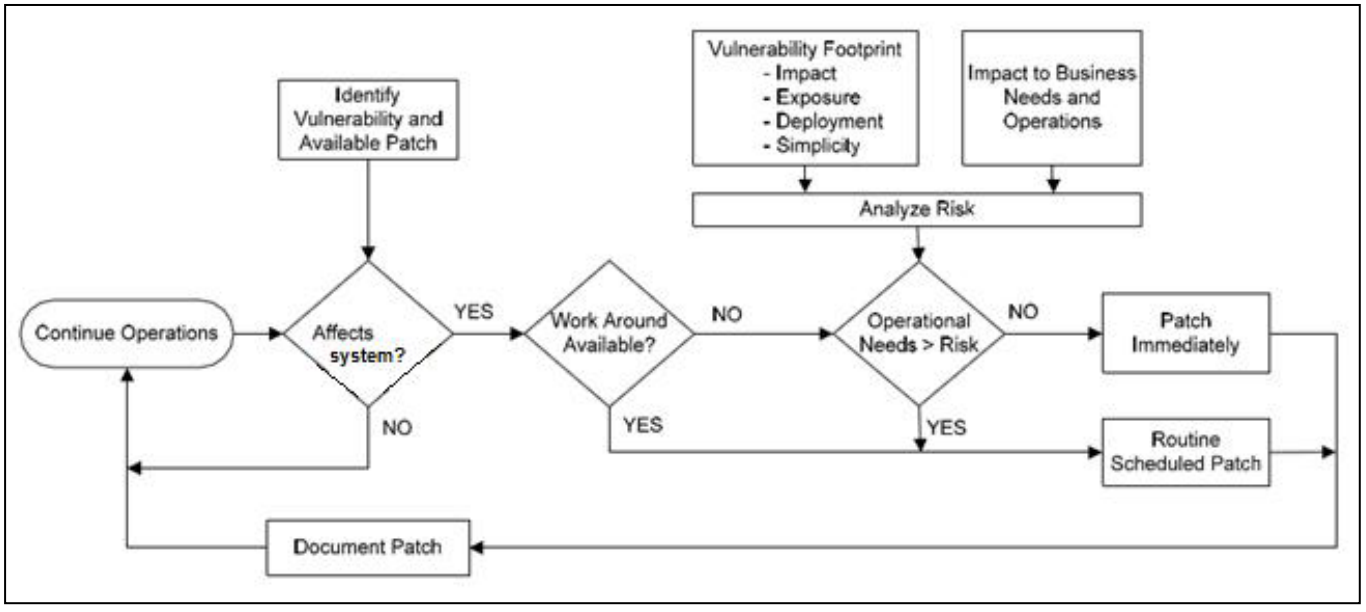

Figure 4 Patch Process

If the urgency determination requires immediate action and a workaround solution is either not available or not the best option, then the following actions should to be taken:

1. Where possible, create a backup/archive and verify its integrity by deploying it on a standby system.

2. Create a checklist/procedure for patch activities and deploy the patch on the standby system.
3. Test the patched standby system for operational functionality and compatibility with other resident applications.
4. Swap the patched standby system into production and keep the previous unpatched production system as a standby for emergency patch regression.
5. Closely monitor the patched system for any issues not identified during testing.
6. Patch the standby system after confidence is established with the production unit.
7. Update software configuration management plan and related records.

Even though it is recommended that a standby system be in place, some sectors may not have a requirement for this system. In such cases, at a minimum, there should be a backup and archive performed that has been tested for restore capability.

If the risk does not require immediate action, patching may be delayed until a tested service level patch is deployed or an alternative workaround is developed. The final patch decision may be to wait until the next update of the system.

Below are a few issues to consider when making the final patch decision:

- Can the patch be deployed at a later date within a routine maintenance window?
- Is there a workaround that would provide adequate protection without patching?
- Does the exploit allow an intruder access other restricted systems?
- What is the impact if the entire system had to be reloaded using disaster recovery backup procedures?
- Does the affected system have to remain in continuous operation?
- Is this a critical system that supports life, health, or finance?
- Are other operational modes (for example, manual) available?

If the internal staff lacks training, experience and expertise in evaluating and deploying patches, using the services of a managed software service provider may be a more cost effective approach. There are several managed software service providers who offer services such as patching, configuring, deploying, and restoring systems.

3.5.1 Unit Patch Process

A vulnerability must be reviewed by IT, IT security, process engineering, operations, and senior management (called the Configuration Control Board or CCB) to determine if there is an immediate need to patch the system. If the decision is made not to patch at this time, patch planning/testing documentation should be maintained to support future patch planning. The information that follows provides an example of the steps to take in an ideal situation. There are other approaches available, depending on the sector requirements, system architecture, operational needs, and type of patching (other types include rolling, sequential, and simultaneous).

3.5.1.1 System Patching

If the Configuration Control Board approves the maintenance window for this activity, proceed with the patch. If an asset owner has a redundant system with units in cold or hot standby status, it is always recommended to patch the cold standby units first.

- **Backup or Standby Units**

  A good practice would be to have one or more completely identical systems located at separate locations cycling between operational, standby, and backup status. If redundant standby units are not available, the next best option is to have a working, stable software backup or archive and a representative test bed available for patch testing. If a test bed is not available, it becomes absolutely essential to have a working backup or archive system in place before any patch activities take place. This archive is the last chance to create a known recovery point of the stable operational environment. In the event that none of the recommended options are available, the alternative is to patch on the operational system, which may be an acceptable risk-based approach to mitigate the vulnerability. The criticality of the system being patched and its downtime tolerance must be carefully considered before patching directly on the production system.

- **Backup Patch**

  In the event of multiple redundant systems, an approved and tested patch should be applied first to the units not in production. For organisations that have multiple

production units, the recommended patch management process is to patch the backup units prior to patching the production or hot standby units. The normal risk management process is to minimize the risk prior to implementation on the production unit.

- **Operational Stability**

  Organisations should establish criteria for benchmarking stability. Based upon this established criteria, the newly patched system must be monitored and evaluated for stable operations. The previous unpatched operational unit should not be patched at this time, serving as an emergency standby unit.

- **Production**

  After the operational criterion is achieved in establishing production stability on the backup system, the organisation is now ready to implement the patch on the production unit. The original production unit should then be patched and tested, now becoming the backup to the operational production system (see Figure 5). The final step is to document and update the configuration management plan to include system modifications and the deployed patch update information.

A sample flow chart identifying patching operations is presented in Figure 5 below.

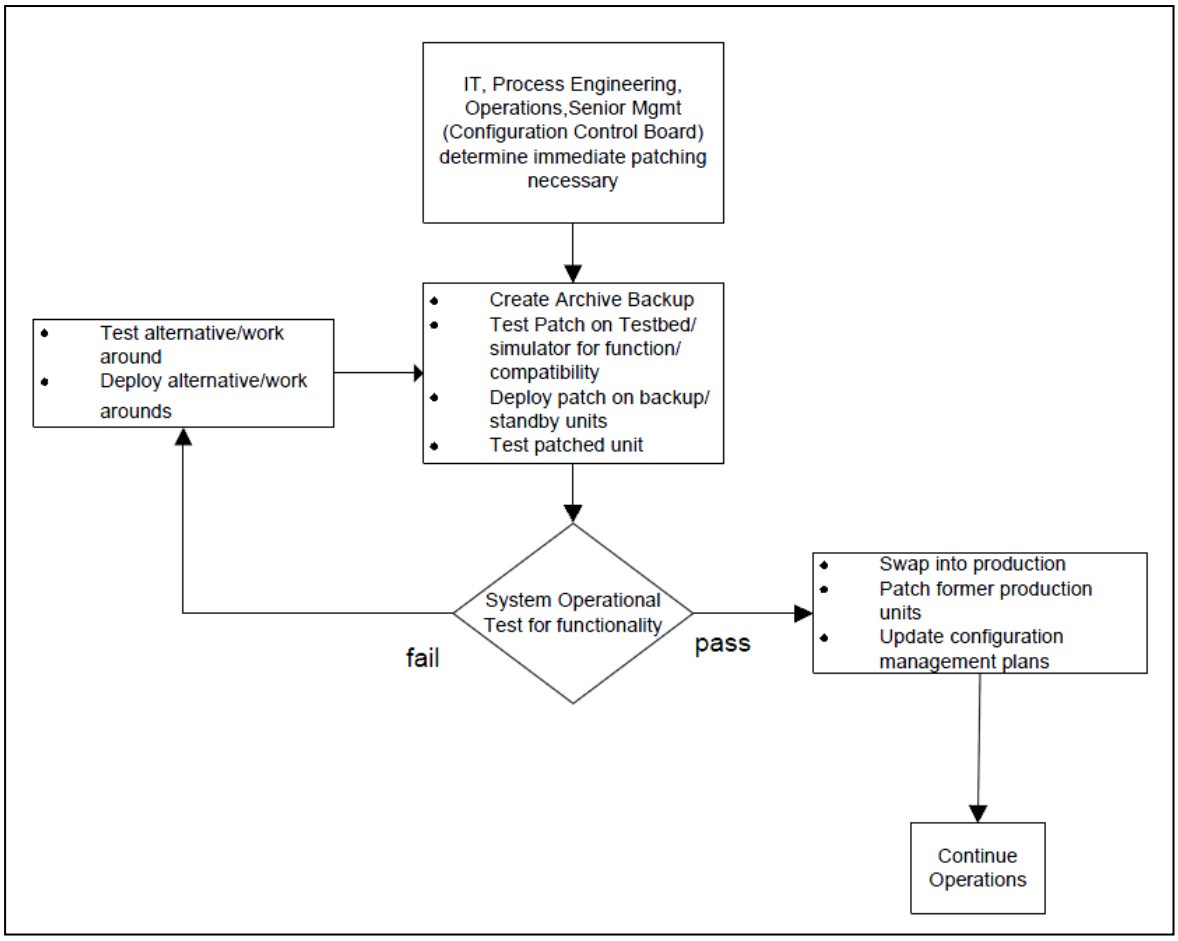

Figure 5 Patching Operations

3.5.1.2 Issues to Consider

The following issues should be considered when creating the patch management plan and the processes and policies related to it:

- **Testing**

  It is always recommended that organisations duplicate the operations environment with absolute functional reliability, but issues associated with component cost and test space may limit the ability of the organisation to have a fully functional test unit. For some scenarios it is adequate to simulate application functions without absolute system replication reliability. The primary function of a test bed/simulator is to mitigate risk prior to implementing changes to the operational environment. An additional benefit from a test bed/simulator is to allow operator training on new

configurations, develop checklists, and evaluate procedures prior to deployment on production systems.

- **Archiving**

  An archive image or data backup of the existing stable operating system should be captured before production patching is conducted to create a valid restoration point. It is recommended that the organisation backup the operational system and restore it on the test bed/simulator system. This activity helps validate that the restore point is usable for disaster recovery. Backup/archiving is frequently done in operations, but the attempt to restore this backup to a working stable environment is frequently only done when needed in a disaster.

- **Rollback**

  Depending upon the patch, a contingency plan should be developed in the event the failure or incomplete patching activities can cause expensive physical damage to equipment. A recommended contingency practice is to create a recovery point by archiving/imaging the current stable system as part of a backup/disaster recovery plan. The organisation would then develop and test uninstalled activities and have hardware spares identified and available (such as power supplies, system motherboards, hard drives, communication switch boards) depending on the criticality of the system.

- **Contingency**

  Organisations should consider the worst case scenario in developing contingencies. Assuming a worst case scenario where patch installation does not restore the system to a stable condition or patch installation and/or removal activity affects other applications; determine if the disaster recovery point restores the system to a stable configuration. An organisation should establish criteria based upon the systems functionality over a specific duration that incorporates timing considerations. It is recommended that organisations know if a patch can be safely and quickly removed and how long this evolution takes as a contingency measure.

A final system operational test plan should be developed to exercise, validate, and document all important identified operational and functional testing points of all primary applications

running in the same environment. This is to ensure stable functional system operations prior to a return to service.

4.0 Conclusion

Any organisation's IT infrastructure is exposed to security risks if it does not have a vulnerability and patch management process in place. A well-defined process provides an organisation with a constant outlook of the vulnerabilities in its IT systems. This allows management to take proper decisions with regards to patching the systems in order to reduce the risks.

5.0 References

- Recommended Practice for Patch Management of Control Systems, US Department of Homeland Security
- Creating a Patch and Vulnerability Management Program, NIST
- Implementing a Vulnerability Management Process, SANS

Appendix A

List of Acronyms

GUI	Graphical User Interface
SaaS	Software as a Service
DoS	Denial-of-Service
CCB	Configuration Control Board