Global Information Assurance Certification Paper

Framework for building a Comprehensive Enterprise Security Patch Management Program

GIAC (G2700) Gold Certification

Author: Michael Hoehl, mmhoehl@gmail.com
Advisor: Kees Leune

Accepted: December 22, 2013

Abstract

Patch Management is an easy concept to understand, but a challenge to execute. With client-side attacks becoming prolific, implementing security updates in a timely manner is becoming even more critical to protect information systems. There are several steps necessary for effective, sustainable patch management including vendor notification tracking, risk assessment, software packaging, and deployment. The purpose of this paper is to present a patch management framework for a typical enterprise based on authoritative standards (e.g., ISO 27002 and NIST) as well as regulatory requirements (e.g., PCI DSS).

"*Provisioning and Access Controls only restrict well intended individuals if product defects and configuration errors persist. I am not well intended.*"

-- Anonymous Hacker

1. Introduction

The concept of a patch is pretty straight forward and broadly understood. In business terms, patching is a form of quality control and defect repair. When a manufacturer identifies and reports a product defect, it is reasonably expected that the consumer (individual and institution) must have the fix applied in a timely manner or accept the associated risk(s). In technology terms, patches are additional code to replace logic flaws in existing software. Consumers have the same obligation to have the fix applied. When the patch is necessary to prevent unauthorized circumvention of a security control, the scope grows from quality control to include risk management. Arguably, an organization that is not effectively managing security patching is not effectively managing quality and risk.

Many organizations have regulatory and legal obligations to implement security updates in a timely manner. For some of these organizations, non-compliance with patching can have a huge impact on their ability to conduct business. For example, American Express has the right to impose non-validation fees on merchants and terminate the Agreement if merchants do not fulfill these requirements (American Express, 2013). Not being able to accept credit cards might shut down the merchant website and severely impact the retail stores. For other organizations, there is business value adopting security authoritative standards. ISO 27002 states, "organization should define and establish the roles and responsibilities associated with technical vulnerability management, including vulnerability monitoring, vulnerability risk assessment, patching, asset tracking and any coordination responsibilities" (ISO, 2013). Customers may prefer to do business with organizations that comply with these best practices.

Author: Michael Hoehl, mmhoehl@gmail.com

The fear of non-compliance, compromised system and data loss are common concerns in today's global environment. Cyber-attacks can originate from within an organization or from the far side of the globe over the Internet. Vulnerabilities impact the effectiveness of design, implementation, and administration of security controls. Patching is so important that CSIS 20 Critical Security Controls includes as number 4: Continuous Vulnerability Assessment and Remediation (CSIS, 2013). Fortunately, a large number of cyber-attacks can be defeated with properly patched computer systems. "According to CERT/CC, about 95 percent of all network intrusions could be avoided by keeping systems up to date with appropriate patches" (GAO, 2003). Without product defects to target, the attacker must depend on human error (e.g., misconfiguration) for successful exploitation.

So, if security patching is so obviously important, why isn't everyone doing it? The first challenge is the sheer volume of security patches. In 2012, 2503 vulnerable products were discovered, with a total of 9,776 vulnerabilities in them (Secunia, 2013). Further, the assumption that patching Microsoft Windows operating system addresses the majority of critical security vulnerabilities is not valid. According to NSSLabs, "…vulnerabilities in the operating system only represent a fraction of the total vulnerabilities of a typical endpoint. Patching the operating system alone is not enough." (NSS Labs, 2013). Today's typical computer has several software applications and utilities that reveal additional surfaces of attack. Just keeping track of all relevant patches for these vulnerabilities and performing initial risk assessment can be a considerable commitment of time and resources.

Deployment of the security updates can also be a challenge. Patch deployment is actually a subset of software deployment. If an organization has a mature framework in place for software lifecycle management including software deployment and removal, then integration of security updates is reasonable. However, when there is no mature software lifecycle management in place, security updates require their own vehicle of delivery.

Author: Michael Hoehl, mmhoehl@gmail.com

Many organizations do not have IT Asset Management or CMDB in place. Trying to determine what systems require the new patch can be a big challenge when there is not a clear understanding of the assets that should be in scope. The first sentence under implementation guidance for ISO 27002 12.6.1 Management of technical vulnerabilities is, "A current and complete inventory of assets is a prerequisite for effective technical vulnerability management" (ISO, 2013). Information such as computer location, software installed, and version of software are all necessary to make intelligent risk management decisions regarding security patching. IT Asset Management provides helpful insight into surface of attack caused by the vulnerability and the impact if a broad exploit is attempted.

Today's workforce is increasingly mobile. According to Global Workplace Analytics, telework growth is up 32% from 2005 thru 2012 (GWA, 2013). Computer assets might not be idle in the office long enough for successful software deployment. In years past, when most employees had a large footprint desktop computer, it was reasonable to expect that patching in the office would be substantially successful. However, many organizations today are issuing laptops and tablets to employees. These employees use the computer frequently; however they might not be in the office frequently. This can create many challenges including updating asset information, determining patch eligibility, patch deployment, and compliance reporting.

Lastly, reporting on the current risk condition can be the biggest challenge. Presenting the impact of a vulnerability in factual, quantitative terms is an effective approach to compel management for action. The problem is gathering the data and creating the risk metrics. Simply reporting the number of unpatched systems is valuable for operations management, but is not adequate for effective risk management. A single critical patch with no known attempts to exploit might be of lower risk than 10 less severe patches with known "exploits in the wild". Location of the asset also has an influence on risk condition. A laptop directly connected to the Internet might be of greater immediate risk than a server with the same vulnerability located within a firewall segmentation and intrusion prevention system.

Author: Michael Hoehl, mmhoehl@gmail.com

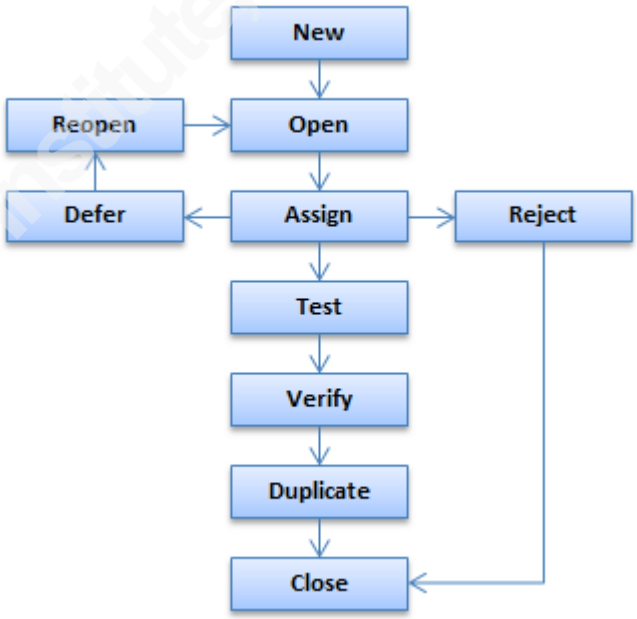
The purpose of this paper is to provide a patch management framework that addresses these challenges in a typical enterprise. The proposed framework includes using automated software deployment solutions to help systematically manage patching.

2. Patch Management Lifecycle

As mentioned earlier, patches are additional code to replace logic flaws in existing software. Defect management is a defined part of the Software Development Life Cycle (SDLC) and considered one of the most important quality control aspects (ISO, 2008). It can occur in any of the SDLC primary processes. Defect Management has a specific internal structure with 10 unique states that align well with a mature patch management lifecycle.

Each state has a relationship with other states as demonstrated below:

Figure 1: Defect Management States



The following provides details about each state and aligns patch management lifecycle events.

Author: Michael Hoehl, mmhoehl@gmail.com

2.1. New

For this state, the vendor announces or customer discovers a new vulnerability. This vulnerability is reviewed by the vendor quality assurance team. Patch and non-patch (e.g., configuration changes) are developed for risk remediation. For common commercial software, the vendor announces the vulnerability and patch. In other cases, a vendor contacts customers directly to notify them of risk and required action. Several third-party service providers provide vendor consolidated notification of vulnerability (e.g., United States Computer Emergency Readiness Team National Cyber Alert System, IBM XForce, Symantec SecurityFocus, SANS @Risk, etc.).

2.2. Open

The vulnerability notification and security patch solution is reviewed by the customer. During this state the Asset Inventory Management and CMDB provide great value in determining if the vulnerable software is present on an information technology asset. Some organizations elect to use network based vulnerability scanners to identify assets that require the patch remediation. Agents installed on assets in scope report back to the vulnerability management console for vulnerability not necessarily revealed using a network scan.

After the presence of the vulnerability and eligibility for the patch has been confirmed, risk assessment begins. ISO 27002 12.6.1.d states, “once a potential technical vulnerability has been identified, the organization should identify the associated risks and the actions to be taken” (ISO, 2013). Several helpful methodologies for risk assessment exist including Factor Analysis of Information Risk (FAIR, 2007), NIST SP800-30 Guide for Conducting Risk Assessments (NIST, 2012), and OCTAVE (SEI, 2001). A metric standard is typically established to uniformly report risk levels across multiple products and vendors. Risk ratings can be acquired by an authoritative source (e.g., MITRE CVE), by the software vendor (e.g., Microsoft), or patch management system (e.g., Lumension or Secunia).

2.3. Assign

Once patch eligibility and initial risk assessment is complete, the asset owner must be approached. The asset owner is the individual within the organization that has ultimate accountability for the asset confidentiality, availability, and integrity. For example, a CFO would be considered the asset owner of a financial management system. During this state, the asset owner is informed of the risk and options to remediate. Discussions regarding timing of remediation implementation are held. Change windows and change freezes are confirmed so that the security update does not result in a service interruption during time of critical business demand. In some cases, the patch might be deferred to a later time or rejected.

IT resource planning occurs during this state. This includes system admins, application support, developers, QA team, and network staff as well as vendor partners. Release management procedures are confirmed. Several actions occur during this phase including acquisition of the patch, preparation of an installation package (e.g., .msi file), distribution of package to regional offices and server repositories, establishing collection of computers that are target for patch package, and initial deployment schedule planning.

Finally, authorization to advance is typically required from the asset owner(s) and data center manager.

2.4. Defer

As mentioned in the Assign state, the data owner or IT may elect to delay the deployment of a security patch. There are many reasons for deferring including avoiding change during retail merchant peak season or delaying until manufacturer retooling of assembly line. Management authorization to defer and formal risk acceptance are formally documented in this state. In many cases, this documentation will be audited.

2.5. Reject

As mentioned in the Assign state, the data owner or IT may elect to reject the deployment of a security patch. Reasons include known system integrity problems

resulting from software update, planned retirement of information asset, or compensating control. Management authorization to reject and formal risk acceptance are formally documented in this state. In many cases, this documentation will be audited.

2.6. Test

This state includes testing and implementation of the security update. This state is the most resource consuming and requires the most coordination. Following Change Management and Release Management policies is vital. Depending on how urgently a security vulnerability requires remediation, implementation should be carried out according to the procedures related to change management or by following information security incident response procedures (ISO, 2013).

In some cases, the security updates do not include a back-out or uninstall option. Proper testing and phased implementation are the best methods for early detection of problems introduced by the new code.

2.7. ReOpen

The purpose of this phase is resume the implementation of a patch after a business decision was made to defer. As with any new patch, the decision to advance the patch into production should be reviewed and approved by the asset owner and IT.

2.8. Verify

This stage confirms the intended patches are in place as intended. Evidence of compliance is typically gathered at this time to demonstrate sustained patch management. In some cases, the system used to deploy the security patch is used to confirm success. When the software change is visible to the employee and part of a common business task, this is reasonable. For example, updating to Microsoft Word 2010 results in user interface changes that an employee can distinguish and confirm successful deployment. Patches and security updates are typically invisible to the employee and cannot be inspected for completion by business staff. In this case, organizations may elect to use a different product to validate patch implementation from the product that implemented

patches. An example of this approach is Microsoft System Center for patch implementation and Secunia CSI for validation. Network-based vulnerability scanners are also popular for enterprises to validate patches are in place and vulnerabilities have been remediated. Examples, of these products include Qualys QualysGuard and GFI LANGuard.

2.9. Duplicate

A vulnerability might manifest multiple times in multiple locations. Subsequent installations of software might reintroduce the vulnerability. The best example of this is Adobe Acrobat Reader. Many business applications include an installation of Adobe Acrobat Reader for presentation of installation and support documentation. This state is intended to identify these duplicate vulnerabilities and trigger installation of relevant patches as part of the application program installation.

2.10. Close

The Configuration Management Database (CMDB) and Asset Management Databases are updated at this time. All as-built documentation and runbooks are also updated reflecting the security updates. Change tickets are closes, Change Management quality review is triggered, and lessons learned documented.

Table 1 below provides a summary relating defect and patch management states:

Table 1: Defect and Patch Management States

State	Defect	Patch Management
1	New	New vulnerability and patch announced by vendor
2	Open	Confirm vulnerability and perform risk assessment
3	Assign	Engage asset owner and assign IT custodian(s)
4	Defer	Accept risk and schedule patch for future implementation
5	Reject	Accept risk and do not deploy patch
6	Test	Test and implement patch
7	ReOpen	Resume deferred patches
8	Duplicate	Eliminate duplicate vulnerabilities
9	Verify	Formally verify patch is in place
10	Close	Update CMDB/as-built documentation. Lessons learned

In summary, the 10 states of SDLC defect management are very similar to a mature patch management program. Organizations that have adopted SDLC will find patch management discussions familiar when relatating to these same 10 states.

3. Patch Management Framework

Now that the lifecycle of patch management has been reviewed, the next step is to identify the appropriate framework to advance patching within a typical enterprise. The National Institute of Standards and Technology (NIST) provides useful guidance with Special Publication 800-40 v2.*0 Creating a Patch and Vulnerability Management Program* and Special Publication 800-40 v3 *Guide to Enterprise Patch Management Technologies (Draft)*. The recommendations provided by both of these publishing are vendor product agnostic, however the recommendations are relevant to all organizations attempting to implement a patch management framework. The following is a summary of the NIST recommendations (NIST, 2005):

- ✓ Create a patch and vulnerability group (PVG) to facilitate the identification and distribution of patches within the organization
- ✓ Use automated patch management tools to expedite the distribution of patches to systems.
- ✓ Deploy enterprise patch management tools using a phased approach.
- ✓ Assess and mitigate the risks associated with deploying enterprise patch management tools.
- ✓ Consider using standardized configurations for IT resources.
- ✓ Measure the effectiveness of the patch and vulnerability management program in a consistent manner and apply corrective actions as necessary.

The following describes key components of a framework for patch management based on the NIST recommendations.

3.1. Prerequisites for Patch Management success

For all organizations, the first step to a successful patch management program is a patch policy. Typically this policy is part of the Information Security Management

System Policy. The policy must align with business objectives and provides the authority to advance security patching. The policy should contain a few key components to be effective. The scope of what must be patched must be clearly described. Scope can be determined by data classification, asset value, location, and business purpose. Establishing prioritization and timing targets are vital for determining when security updates are to be in place. Procedures for obtaining exemption and who can authorize the exemption (and ultimately accept risk) must be clear. Some form of risk register should be referenced to track the exemptions including authorization and expiration. Ideally, the policy references risk management policies and practices.

Asset Inventory Management is another essential prerequisite for patch and vulnerability management. Before a computer system is accredited or initially commissioned into production, an inventory of software assets installed should be taken. This inventory should be regularly updated. For this reason, manual inspection is not practical. Some form of organization-wide automated scanning is necessary to gather information about the installed program and binary files (e.g., for Microsoft Windows this includes .exe, .dll. and .ocx. files). Several commercial and open source products provide this function including Microsoft System Center, IBM Tivoli, Secunia CSI, and OCS Inventory NG. As mentioned earlier, this software is critical for determining eligibility for a patch. Without asset inventory information, it must be assumed that all computer assets need the patch. This results in unnecessary assumption of risk and consumption of infrastructure resources (e.g., system processor and storage, network capacity, etc.).

It is sometimes helpful to add metadata to the inventory database that cannot be harvested directly from the asset. Information such as geographic location, data classification, and redundancy is valuable when performing initial risk assessment. Details about compensating controls (e.g., Host Intrusion Prevention) might also be valuable when stored within the asset inventory. For example, the risk assessment for a single patch might be different for a desktop PC used for word processing located in the Corporate Offices behind a firewall and IPS as compared to a laptop containing confidential information connected directly to the Internet. Many asset inventory

management solutions allow this additional metadata and the ability to create useful tags for associating similar assets.

3.2. Patch and Vulnerability Group (PVG)

NIST SP 800-40 introduces the concept of a Patch and Vulnerability Group. The PVG is a multi-discipline team of individuals with a common mission to manage risk by advancing necessary security patches. According to NIST, the duties of the PVG include the following (NIST, 2008):

1. Inventory the organization's IT resources to identify the hardware equipment, operating systems, and software applications that are used within the organization.

2. Monitor security sources for vulnerability announcements, patch and non-patch methods of remediation, and emerging threats that match up with the software within the system inventory of the PVG.

3. Prioritize the order in which the organization addresses the remediation of vulnerabilities, based on analysis of risks to systems.

4. Create a database of remediation methods that need to be applied within the organization.

5. Conduct the testing of patches and non-patch remediation methods on IT devices that use standardized configurations.

6. Oversee the vulnerability remediation process in the organization.

7. Distribute vulnerability and remediation information to local administrators.

8. Perform automated deployment of patches to IT devices using enterprise patch management tools.

9. Configure automatic updates of applications whenever possible and appropriate.

10. Verify vulnerability remediation through network and host vulnerability scanning.

11. Train administrators on how to apply vulnerability remediation.

The PVG team membership is intentionally diverse. Members include representatives from IT, security, key business functions, and management. The size and structure of the PVG varies according to organization complexity. The PVG approach provides many

benefits including multiple disciplines, subject matter expertise, business awareness, and resource management. Ultimately, the PVG provides the key risk management guidance and authority necessary to advance the patch remediations.

Appendix A: Patch Management Workflow using PVG includes a swim lane flow chart showing a patch management process involving the PVG and other relevant roles.

3.3. Tools and Automation

The aforementioned NIST guides emphasize the use of automated tools for sustainable patch management. Widespread manual patching is no longer effective for risk and resource management as the number of patches necessary for vulnerabilities grows and threats continue to rise. Key functions for patch management framework automation include:

Table 2: Patch Management Function and Examples of Automation

Patch Management Function	Examples of Automation
Vendor notification tracking	IBM XForce, Symantec SecurityFocus, Secunia VIM, SANS @Risk, US-CERT National Cyber Alert System
Asset inventory management	IBM Tivoli, Lumension, Microsoft System Center, OCS Inventory NG, Secunia CSI, Symantec Alteris
Vulnerability detection and patch eligibility	IBM Endpoint Manager, Lumension, McAfee, Microsoft System Center, Secunia CSI, Symantec Alteris
Risk assessment	Secunia CSI, Lumension, McAfee, Qualys
Software packaging and deployment	IBM Endpoint Manager, LANDesk Microsoft System Center, Secunia CSI, Symantec Alteris

Note! This table is not intended to be an exhaustive listing of product options, and not intended to serve as a product endorsement.

At this time, there is no single solution that provides the necessary automation for all technology requiring patching. Many exceptional tools are available open source and commercially, however the products tend to favor a specific environment (e.g., Microsoft Windows) or purpose. More than one solution is most likely required for a medium or large enterprise. When implementing the tools, a phased approach is recommended. This allows for integration into the organization and ultimately better adoption.

As the automated tools accumulate information about vulnerabilities, essentially a "recipe box" is being created to successfully and substantially exploit the organization's

technology assets. Care should be taken to implement a secure design and regularly examine the security controls associated with the new patch management tools so that additional unintended risk is not created (don't give your recipe away). Logical Access Controls, Least Privilege and possible Segregation of Duties should be considered, too.

3.4. Remediation Database

With this proposed patch management framework, all requests for security update exemption must be presented to the PVG for consideration. This includes permanent and temporary exemption. In some cases, a security control may be in place that adequately protects against attempts to exploit a known vulnerability. For example, buffer overflow attacks may be prevented by a host intrusion prevention system. Consideration for a compensating control must also be presented to the PVG for consideration. Ideally, a security update exemption form must be completed prior to presentation to the PVG. This helps the PVG fully understand risks and ramifications. All PVG approved exemptions are then formally presented to an IT Director and the Asset Owner for risk acceptance and authorization. PVG will maintain the authoritative record of risk acceptance history.

Tracking the compensating controls and exemptions over time can be overwhelming without some form of risk register. The purpose of the risk register (also known as Remediation Database in NIST 800-40v2) is to track remediations that need to be applied to assets with the organization. The Remediation Database is extremely valuable when audits are performed (e.g., SOX ITGC, PCI, etc.). Ideally, the Remediation Database is part of the Asset Inventory Management or Risk Assessment information system.

3.5. Metrics

Metrics are necessary to demonstrate the effectiveness of the patch management program and current vulnerability condition. Performance measurement is typically used for assurance, but also can be used to motivate change and reward effort. There are many

types of patch and vulnerability metrics. NIST includes recommendations in SP 800-40 v2. Essentially, there are 4 general types listed below with example metrics:

Table 3: Example Patch Management Metrics

Metric Type	Metric Examples
Patch Program Maturity	Response time for accepting vulnerability notification Response time for risk assessment Response time for testing Change Management violations Change Management changes/reschedules Patch packaging duration and level of effort
Compliance	Patch infrastructure readiness to patch Planned patches in place
Risk	Susceptibility to attack Duration of patch delivery Number of exemptions Planned patches not in place New computers missing patches Emergency patching Unpatched, unauthorized software missing patches
Cost	Cost of PVG Cost of tools Cost of services Cost of rework or redeployment

Note! This table is only a sample of key metrics and not intended to be an exhaustive listing of all patch management metrics

Organizations are not advised to attempt creating all metrics up front. Metric development is an iterative process that evolves thru many levels of maturity. Initially, the Information Security Management System Policies should be mapped to the metrics to demonstrate policy compliance. This helps avoid the "so what?" response to metrics. Ideally, IT custodians should report metrics to Functional Manager, Asset Owner, Security, and PVG within 30 days of vendor update release. Recurring patch performance reporting is necessary to credibly demonstrate sustained safeguarding of software and data assets. Reports might need to be archived to demonstrate the patch program has been sustained and patching remains compliant.

4. Keys to Success with Patch Management

There are many reasons for patch management program failure. These include:

- No Corporate policy requiring patching

- No clear understanding of roles and responsibilities associated with patching
- Wrong expectations of scope
- Poor software lifecycle management (EOL software not removed, multiple releases of same software installed with different versions, etc.)
- Attempting to use one solution for all needs
- No release or change management maturity
- No tools or automation to support process in a repeatable manner
- No computer build standard or accreditation for new computers

There are a few keys to success when considering the implementation of an Enterprise Patch Management Program. Communication within the organization before, during, and after patching is vital. ISO 27002 Section 16.2.1.a advises, "the organization should define and establish the roles and responsibilities associated with technical vulnerability management, including vulnerability monitoring, vulnerability risk assessment, patching, asset tracking and any coordination responsibilities required" (ISO, 2013). Consider the use of a RACI chart to clarify roles and responsibilities. This helps prevent communication breakdown because of gaps or uncertainty with patch management duties.

Variance in deployed versions of software makes patch management more challenging. One of the best ways to be successful is to patch less with the use of software configuration and version standards. Patch management benefits from computer build standards and accreditation include less vendor patch notifications to track, less software updates to package, reduced demand on the network to deploy software updates, and less variety of validation to perform. Software configuration management and software lifecycle management go hand-in-hand. Elimination of outdated and end-of-life software reduces the surface of attack. Further, a new version of software may offer improved native security features that defeat attacks. In the Microsoft Security Intelligence Report in 2013, computers running Windows XP in the first six months of 2013 encountered about 31 percent more malware worldwide than computers running Windows 8, but their infection rate was more than 5 times as high (Microsoft, 2013). Of course, standardization can be quickly intentionally and unintentionally be undone when least privilege is not applied. To prevent this from occurring, ISO 27002 16.2 advises

"…organization should define and enforce strict policy on which types of software users may install" (ISO, 2013). A small investment in the planned build process, software lifecycle management, and restriction on software installation can result in long term time savings by reducing variance and ultimately the amount of patching.

A successful patch program grows iteratively. Seldom is such a program successful by implementing the new framework and procedures all at once. As the patch management program matures, use metrics to track growth, identify key areas needing management attention, and celebrate improvement.

5. Conclusion

Patch and Vulnerability Management remain one of the top requirements for a successful security program. In the 2013 analysis brief on vulnerability threat trends NSS Labs advises, "Implement effective patch management programs wherever possible. Vulnerabilities in software will continue to be a major risk factor, increasing the importance of patch management in the critical path to security" (NSS Labs, 2013). A mature patch management lifecycle is very similar to the SDLC Defect Management states. These states can be properly managed using best practices like ISO 27002 and a framework like that proposed by NIST SP800-40v2. A successful framework includes policy, asset inventory control, risk management, standardization, and metrics. Fortunately, today's enterprise has many tools available to choose from that will automate key aspects of a patch management program. This automation based on a solid framework will assure patch management success for today's enterprise.

6. References

American Express. (2013). *Data Security for Merchants Compliance Requirements*. Retrieved from https://www260.americanexpress.com/merchant/singlevoice/dsw/FrontServlet?request_type=dsw&pg_nm=merchinfo&ln=en&frm=US&tabbed=complienceRequirement

Author: Michael Hoehl, mmhoehl@gmail.com

CSIS. (2013). *20 Critical Security Controls - Version 4.1*. Retrieved from http://www.sans.org/critical-security-controls/guidelines.php

Davis, Noopur. (2013). *Secure Software Development Life Cycle Processes*. Retrieved from https://buildsecurityin.us-cert.gov/articles/knowledge/sdlc-process/secure-software-development-life-cycle-processes

FAIR. (2007). *An Introduction to Factor Analysis of Information Risk (FAIR).* Retrieved from http://riskmanagementinsight.com/media/documents/FAIR_Introduction.pdf

GAO. (2003). *Effective Patch Management is Critical to Mitigating Software Vulnerabilities.* Retrieved from http://www.globalsecurity.org/security/library/report/gao/d031138t.pdf

GWA. (2013). *Latest Telecommuting Statistics.* Retrieved from http://www.globalworkplaceanalytics.com/telecommuting-statistics

ISO. (2008). *ISO/IEC 12207 for Software Life Cycle Processes*. Retrieved from http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=43447

ISO. (2013). *ISO/IEC 27002:2013 Information technology -- Security techniques -- Code of practice for information security controls.* Retrieved from http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=54533

Lipner, Steve & Howard, Michael. (2005). *The Trustworthy Computing Security Development Lifecycle.* Retrieved from http://msdn.microsoft.com/en-us/library/ms995349.aspx

Microsoft. (2013). *Microsoft Security Intelligence Report, Volume 15, January through June, 2013.* Retrieved from http://www.microsoft.com/security/sir/default.aspx

Nicastro, Felicia. (2011). *Security Patch Management.* Boca Raton, FL: CRC Press

NIST. (2005). Special Publication 800-40 v2.0 (Draft) Creating A Patch and Vulnerability Management Program. Retrieved from http://csrc.nist.gov/publications/nistpubs/800-40-Ver2/SP800-40v2.pdf

NIST. (2010). *Special Publication 800-37 Guide for Applying the Risk Management Framework to Federal Information Systems.* Retrieved from http://csrc.nist.gov/publications/nistpubs/800-37-rev1/sp800-37-rev1-final.pdf

Author: Michael Hoehl, mmhoehl@gmail.com

NIST. (2012). *Special Publication 800-30 Guide for Conducting Risk Assessments* Retrieved from http://csrc.nist.gov/publications/nistpubs/800-30-rev1/sp800_30_r1.pdf

NIST (2013). *Special Publication 800-40 v3.0 Guide to Enterprise Patch Management Technologies (Draft).* Retrieved from http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-40r3.pdf

NSS Labs. (2013). *Vulnerability Threat Trends; A Decade in Review, Transition on the Way.* Retrieved from https://www.nsslabs.com/reports/vulnerability-threat-trends

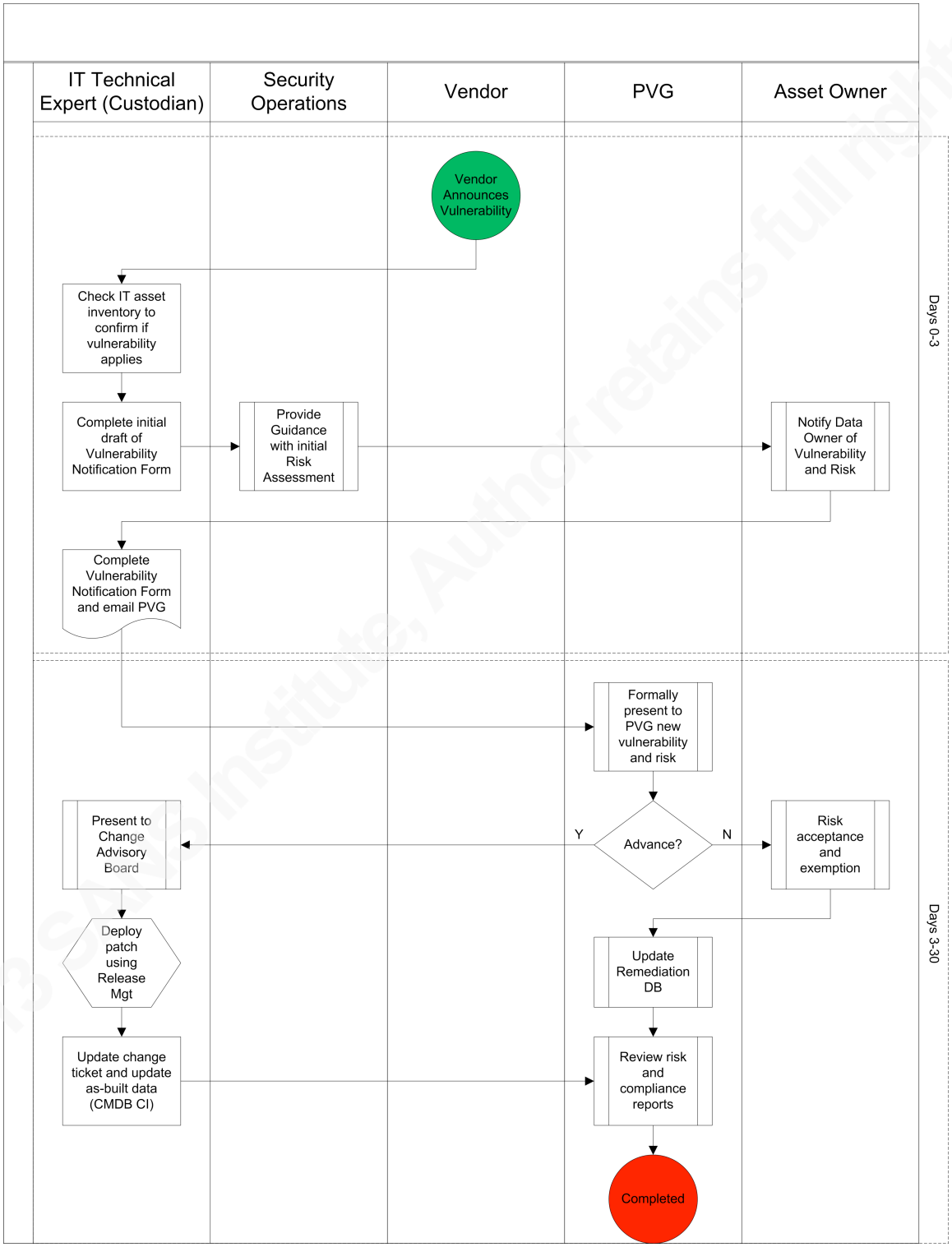
RMI. (2006). *An Introduction to Factor Analysis of Information Risk (FAIR).* Retrieved from http://www.riskmanagementinsight.com/media/docs/FAIR_introduction.pdf

SEI. (2001). *OCTAVE method*. Retrieved from http://www.cert.org/octave/octavemethod.html

Secunia. (2013). *Secunia CSI 7.0 – Technical User Guide.* Retrieved from http://secunia.com/?action=fetch&filename=Secunia_CSI_7.0_Technical_User_Guide.pdf

Secunia. (2013). *Secunia Vulnerability Review 2013*. Retrieved from http://secunia.com/vulnerability-review/

APPENDIX A: Patch Management Workflow using PVG



Author: Michael Hoehl, mmhoehl@gmail.com

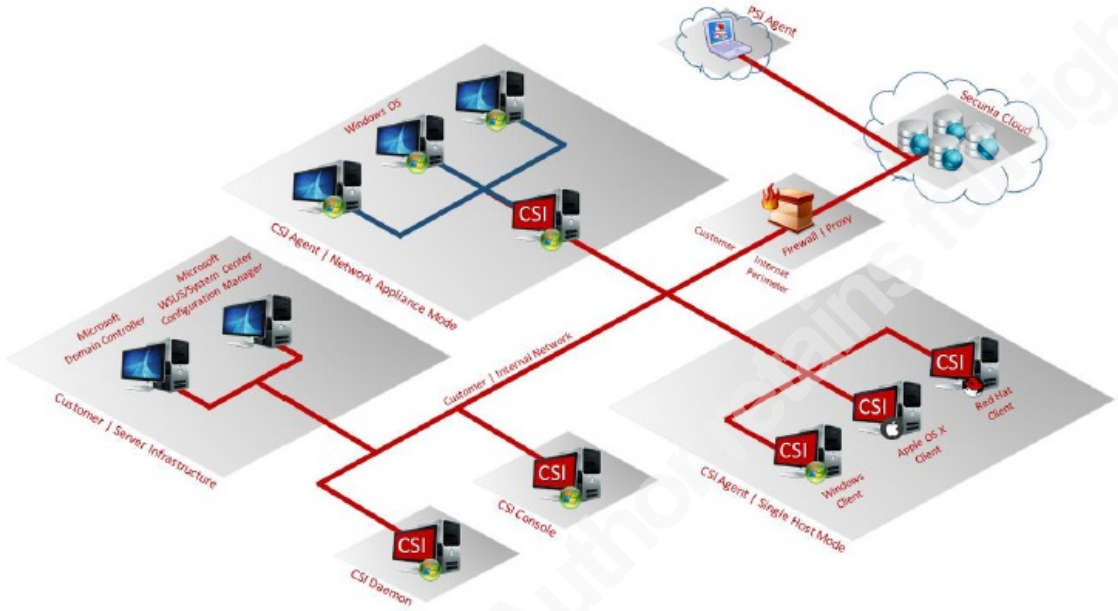
Appendix B: Case Study - Secunia CSI

Corporate Software Inspector (CSI) is a cloud based service offering from Secunia (Secunia, 2013). It offers a combination of vulnerability intelligence, vulnerability scanning, patch creation and patch deployment. In this section, we will demonstrate how the CSI tool and automation can perform patch management functions for an enterprise.

A master database of file signature metadata for over 20,000 application programs and plug-ins is hosted at Secunia. It is one of the largest in the world. The master database contains metadata associated with the .exe, .dll. and .ocx binary files of Microsoft Windows based application programs and plugins. File signature metadata is generic non-sensitive text strings embedded in the aforementioned binary files. In addition to Microsoft Windows the Mac OSX, Red Hat Enterprise Linux, and Android environments are also support.

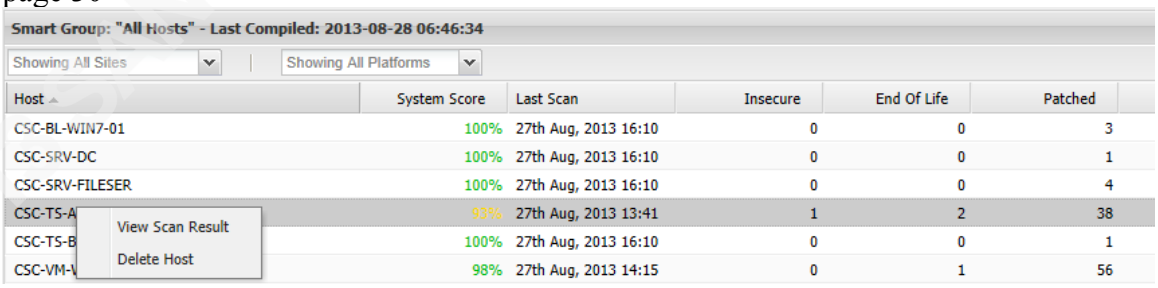
As mentioned earlier, one of the key pre-requisites of a successful patch management program is asset inventory management. Secunia has a highly effective approach to create this asset inventory. A web browser is used by the customer to access the CSI console in the cloud and initiate an authenticated scan of a target computer. Using the Secunia CSI agent (or agentless approach with the help of Microsoft System Center) performs the scan and securely transfers the metadata back to Secunia over the web. All installed programs and plug-ins are identified when the metadata harvested from the target computer is compared with the Secunia master database. The inventory results are then presented for review using a web browser. The entire Secunia CSI solution uses a very small footprint at the customer location. The following provides a visual representation of all the various type of scans and the integration with the Secunia cloud infrastructure:

Figure 1: Secunia scan types and integration with Cloud Infrastructure from Secunia CSI 7.0 – Technical User Guide page 12

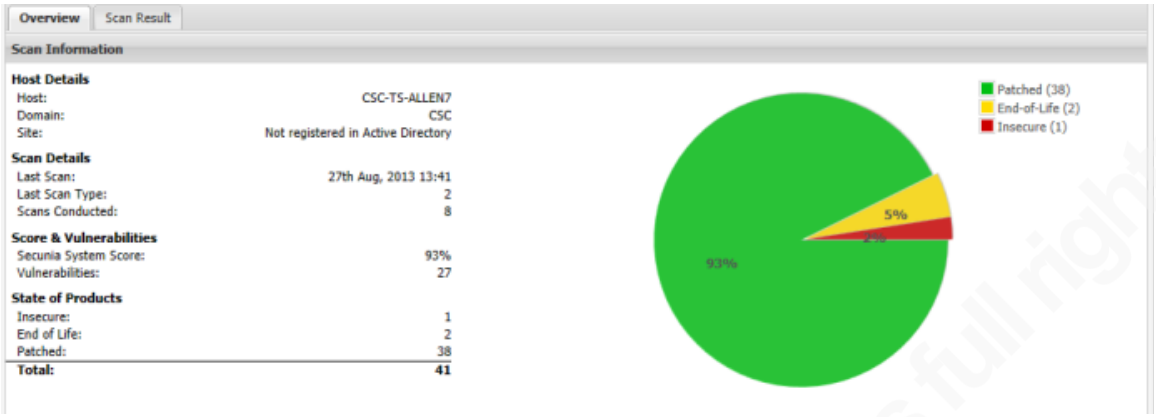


The asset inventory is then correlated with vulnerability information based on Secunia Vulnerability Intelligence. The results of this correlation include identification of eligible security patches (state) and system score (risk). An example of the web console including scan results and system score are provided below:

Figure 2: Secunia web console example from Secunia CSI 7.0 – Technical User Guide page 30



Author: Michael Hoehl, mmhoehl@gmail.com



This system score shown above provides valuable insight during risk assessment. This is very helpful when managing patch cycles that include multiple patches from multiple vendors. Because of the cloud architecture, the Secunia CSI agent can gather necessary information from computer assets wherever they are located on the Internet. Mobile computers do not need to return to the office or connect with Corporate VPN to report in for patch eligibility and risk assessment.

Asset inventory, patch eligibility and risk assessment are some of the key challenges for getting a patch management program started. With Secunia CSI as part of the patch management framework, these can be accomplished quickly and securely.

Secunia also offers security patches prepackaged for installation using Microsoft System Center and similar endpoint management solutions. The catalog of patches is quite extensive. The native tool from Microsoft (Update Publisher) is typically used to prepare the installation files from other patch content catalogs. To streamline the software packaging process, Secunia offers an easy to use, 4-step wizard-driven interface to create and customize the packages for the target systems. This is very convenient for enterprises that do not have dedicated software packaging subject matter experts.

There are a number of other features with Securnia CSI that help with patch management. This includes Secunia PSI integration with CSI, agentless integration with SCCM, network scanning, and custom program vulnerability assessment.

Secunia PSI is the personal version of CSI. It was originally intended for individuals at home and non-commercial use. It can now be used for commercial

purposes, too. The scan engine architecture is the same and there is integration with Secunia CSI. In simplest terms, PSI is a mini-CSI with a GUI. There are 2 use cases for PSI that are particularly beneficial for today's enterprise. The first is PSI allows IT staff with high authority (system admins, DBAs, developers, etc.) to self-administer updates to programs that are not part of the general employee standard. Commonly, IT staff must install special utilities, consoles, and element managers to perform their duties. These programs are not the typical business application and not included in the standard suite of supported products. Patching still remains an important requirement—especially with IT staff having high authority to key business systems. The second use case is mobile staff. When there is a large gap in time between visits to the office, laptop computers can quickly fall out of compliance. PSI allows access to all the Secunia prepared patches from their cloud infrastructure over the Internet. By combining PSI and CSI, conscientious IT staff and mobile staff can fast-track patch management independently. The only downside of PSI at this time is no support for custom patches. For this CSI with System Center is required.

For IT shops hesitant to install yet another agent on the computer, Secunia CSI offers an agentless solution with the help of Microsoft System Center. Once Microsoft System Center has been configured for software inventory, Secunia CSI can query the Microsoft SQL Server to harvest the necessary file signature metadata. A network based scanner option is also available for enterprises that do not have System Center in place in a specific location or at all.

Customers can elect to send Secunia the binaries of customer programs to add metadata into the master database. This is especially helpful for applications like hospital lab or radiology equipment, surveillance and building access systems, and Supervisory Control and Data Acquisition (SCADA) devices.