Effective Security Patch Management

ICTN 6823 Information Security Management

Awojana Tolulope

East Carolina University

Abstract

Results have proven over the years that more than 90 percent of successful Internet-based attacks exploit software applications which were inappropriately configured or patched. Many of these attacks are usually conducted by humans or internet based worms. However, with the constant report of vulnerabilities from the various industries across the world there is still a need for improvement in security patch management. Vulnerability Management is an integral part of computer and network security. It is an unending information risk process of managing network security by reducing the damage on the security of a network which occurred as a result of possible design or implementation flaws. Patch management is a subset of vulnerability management. It involves obtaining, testing and installing security patches to reduce or eradicate one or more susceptibilities in a network. This paper would be discussing the process involved in effective security patch management highlighting the events in the life cycle of vulnerability for an organization with the adequate best practices required for an effective security patch management. It would also describe the importance of automating the process to address the increased threats identified by the known security vulnerabilities.

Keywords: Patch Management, Vulnerabilities, Automation, Risk.

Introduction

Patches are one of the most important tools used in cybersecurity required by an everyday technology user. A patch is a small piece of software that is usually issued by a company to correct a problem in cases where a security flaw is detected. (ITRC, n.d.) Patches are constant as no software program is perfect, the more widespread a program is the more chances for problem to take place, hence most of the software programs produced today are usually patched. Software patches are normally released to address the vulnerabilities and irregularities found in software programs. (Fisher, 2018) A good example of a patch that could have been prevented is the Equifax Breach that resulted in the exposure of One Hundred and Forty Three Million (143,000,000) people which was caused by a failure to update the software component of the Apache Struts web-application. The attacker basically exploited the Apache Struts bug and was able to gain access into Equifax's system (Newman, 2017). Another good example of a way hackers exploit a flaw is the WannaCry Ransomware attack that locked down over Two Hundred

Thousand (200,000) computers and networks. This also occurred as a result of a vulnerability discovered in the Microsoft Operating System of which a patch was issued a few weeks earlier before the incident occurred but several users had neither installed or automatically updated their systems. (ITRC, n.d.)

Patch management is not only essential in fixing vulnerabilities related to security, it also assists in warding off business threats. (Sihnoven et. al, 2010) Hence, it has become very important for every business to develop an effective patch management strategy through a proactive approach. Patch management is very vital in the field of cybersecurity as it focuses on the vulnerabilities in software and systems. It is an integral function of network security management. (Cavusolgu et. al, 2008) Effective Patch Management ensures that patches are fielded in a way to eliminate vulnerabilities and prevent threats or attacks in organizations. It introduces an automated discipline that can be adapted to easily once a process is in place with an increase in the level of accountability on roles mainly responsible for security and systems within organizations. "*Patch management is far from simple; it requires careful planning and consideration of factors that could impact the process*." (HCPro, 2018). The rest of the paper is organized as follows. In the second section, we review papers related to this study. In the third section we describe the process involved in effective patch management. In the fourth section we describe the events in the life cycle vulnerabilities and importance of automating the processes. The concluding section summarizes our research.

Literature Review

Gerace et.al (2009) carried out a survey on IT professionals from the different facets of the industry to ascertain the critical elements in the patch management process using the following metrics as a grade point; manual update, windows automatic update and automation. The results indicated 64.4% usage of the automated patch management software, 18.2% usage of the windows automatic update and 16.5% usage of the manual application of patches. It was discovered that a major percentage of the respondents used the automated patch management in their respective organizations. The metrics were further broken down into different factors which includes: identification of vulnerabilities, network scan pre-deployment, dedicated resources, network scan post-deployment, technology inventory maintenance, testing prior to deployment

and senior executive support. Survey results showed that most of the respondents placed very little importance on the senior executive support which implies that most of the patch management implementation are done with little or no supervision from the management. Even though automated patch management tools were commonly used it was suggested that users be included in the patch management process for maximum efficiency.

Palumbo (2015) implemented the Secunia CSI application in Lehigh University using the patch management process for retrieving and testing patches within the university systems, detecting the vulnerabilities and installing on the intended target. Going through almost the same metrics used by Gerace et. al (2009) the number of attack vectors in the Lehigh University system was found to have considerably decreased. Hence, it is essential to have a product in place to manage the updates and evaluate the performance from time to time. Sihnoven et. al, (2010) focused on practical challenges associated with patch management and release management processes and the similar features highlighted from the customers and provider point of views. The data used in this study was collected from an Information Technology (IT) department using the IT Infrastructure Library (ITIL) as their framework. An IT customer and IT company were used as the case study. The data collected from the respondents were analyzed based on the following factors; concepts and classification, roles and responsibilities, process interdependencies and the flow of information. Based on the data collected and analyzed, it was concluded that for an effective patch management process there should be a defined concept for the patches and proper documentation for easy understanding. The policies and metrics for the patches also have to be established. Adequate testing with sufficient resources should also be carried out. Mohammadi (2014) also designed a systems engineering (SE) framework for implementing a security and critical patch management process (SCPMP) in diverse environments. The framework was proposed to minimize the costs of time of operation in IT and reduce the risk of security and vulnerability attacks. The results of the systems engineering (SE) framework tested prove a significant amount of reduction of IT costs in performing the patching process. The patch deployment structure proposed would also assist Information Technology (IT) managers in assigning duties to those responsible for implementing and completing the task process. A detailed review of the various research papers indicates the need for a well-defined patch management process with distinct automated features.

Patch Management Process

A patch management process is an activity that comprises of different critical elements that work hand in hand for the success of the process. (Gerace et. al, 2009) An effective patch management process comprises of the four basic phases dependent on the format of the organization. (Tejaswi, n.d.)

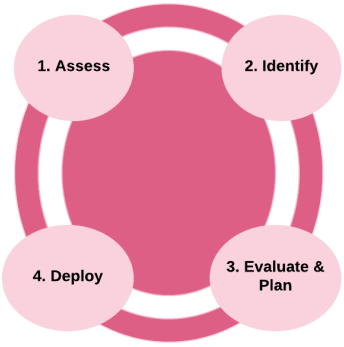

Fig. 1: Fundamental Phases of Patch Management Process

(i) **Assessment Phase:** This phases entails evaluating the security and vulnerability threats that is likely to be encountered by your organization. It requires an understanding of the functions of the hardware and software assets used in the organization. This phase also requires a knowledge of how to protect the assets and the level of the existing patch in the system. It is also important to evaluate the current software distribution infrastructure present in the system, and ascertain the best source of information for software updates. (Tejaswi, n.d.)

(ii) **Identification Phase:** With adequate knowledge of the organization's environment, the identification phase helps to discover new invention of software updates in an authentic way that is relevant to the system. This phase also determines if a software update needs an emergency deployment or a normal process.

(iii) **Evaluation and Planning Phase:** After a patch has been identified, it is important to evaluate the significance of releasing a patch. The classifications and deployment timeframe varies based on the needs of the organization. In the evaluation of a patch, all of the issues, necessities and functionality changes must be properly documented.

It includes a go/no-go decision and an evaluation of what is needed to install the software updates. It also requires testing the software updates to determine the compromise on business-critical systems and applications. (New Boundary, n.d.)

(iv) **Deployment Phase:** After the testing has been carried out and the patch has been certified ready, then it can be deployed. However, it is important that there is a communication to the end users with a walkthrough on the steps to follow for installation. In the event of a problem identified during installation, a post-implementation review should be conducted for proper identification of the weaknesses in the process and lessons learned for improvement. (Barreiro, 2011)

Gerace et. al (2009) also identified the following critical elements that make up a patch management process includes:

- **Senior Executive Support**: This comprises of the senior management in an organization. For a successful patch management, the support of the senior management is required. It involves the identification of information security risk and adequate support from the top on the patch management process.
- **Dedicated resources and clear-cut responsibilities:** This element ensures that each staff properly define, implement and manage their duties in the process.
- **Creating and sustaining a current technology inventory:** Technology inventory of the hardware and software is very essential to any patch management process. It assist the department responsible for patch management to keep record of the vulnerable systems and number of patches that would be required. It can also help the staff to determine the location of computers and the owners.
- **Identification of vulnerabilities and patches:** With the technology inventory, the department in charge can track the systems for vulnerabilities and software patches used in the organization.
- **Scanning and monitoring the network:** This element can be used to analyze the risk level through the software tools and remedial steps taken when necessary.
- **Pre-deployment testing of patches:** This is a pro-active step that should be taken by any organization. It is important that the patches are tested before deployment so as to ensure

they function as proposed and in cases of failed patches adequate measures should be taken to fix the bugs.

- **Post-deployment scanning and monitoring**: To ensure that the patches have been effectively applied, it is very crucial to inspect the network. It could also serve as an audit tool for compliance with the established standards.

Vulnerability Management Life Cycle

According to Threat Analysis Group (n.d.), a vulnerability is a weakness in a security program that can be harnessed by threats to obtain unapproved access to a system. While patch management is important, it is also imperative to understand the life cycle of a security vulnerability. Through the vulnerability life cycle, organizations can discover the weakness in computer systems, rank assets in their order of priority, remediate weaknesses and ensure the vulnerabilities have been eradicated. (CDC, n.d.)

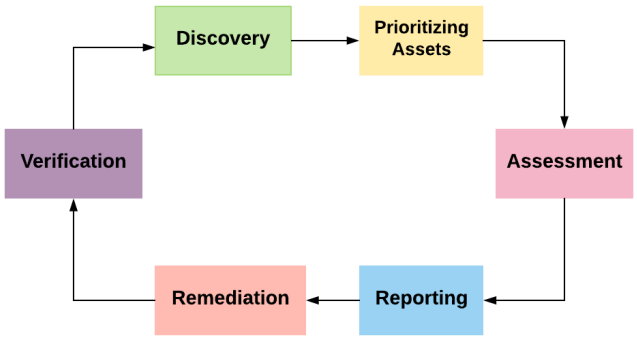

Fig. 2: Vulnerability Management Life Cycle

(i) **Discovery**: This is the first stage of the vulnerability management life cycle. It includes taking an inventory of all the assets to properly identify the operating system and open services. Once a vulnerability is identified, a network baseline can be developed. (CDC, n.d.)

(ii) **Prioritizing Assets**: After a vulnerability has been discovered, this stage helps to prioritize the fixes as issues that pose greater risk will be handled with a higher priority than the ones with minimal risk. (RedHat, 2015) It requires that the assets are grouped based on the level of importance to the business.

(iii) **Assessment**: In this stage, a baseline risk profile is established to mitigate risks based on the criticality of assets, vulnerability threats and asset classification.

(iv) **Reporting**: Here the extent of business risk is measured with the assets in accordance to the security guidelines. It requires that the security plan and framework is documented, highlighting and describing any suspicious and identified vulnerability.

(v) **Remediation**: In this phase it is necessary to prioritize and remediate vulnerabilities according to the level of importance on the business risk. The workaround applied should help deflect the threat and establish controls.

(vi) **Verification**: This final phase requires that the eliminated threats are verified through follow up inspections. There are different methods used for verification some examples of these include; ticketing systems and scanners. (Flexera, n.d.)

Although patch and vulnerability management sound similar but there are some differences noticed. Patch management entails the use of patches, updates and fixes of software that are installed for a variety of reasons. The patches require adequate planning before they are rolled out so as to know the patch required for every system at any point in time. However, vulnerability management is only concerned with software security issues and often times the issues can be resolved with the application of a patch. Patch management is a subset of vulnerability management. Since vulnerabilities exist in all forms of systems, several means can be deployed to eliminate them.

Automating the Patch Management Process

With the plethora of operating systems in existence ranging from Windows Operating Systems to Linux Operating Systems, constantly installing patch updates might be a complex task mostly for administrators and information technology (IT) managers. (Mathivanan, 2017) Fifty percent of Information Technology Professionals surveyed in a Patch Management Study affirmed that the bulk of security patches could be really overwhelming and high to keep pace with. (Chapple, 2017) With the increase in the rate of cybercrime, hundreds of patches are released each month, for example Microsoft's Patch Tuesday is released on the second Tuesday of each month (Fisher, 2018). At the other end, the risk of leaving a system unpatched is dangerous as it becomes vulnerable to many attacks. To better protect systems, it has become important for IT

administrators to look into the automated process. Comodo One (n.d.) highlighted the below reasons why automated patch management is important.

- **Security:** With the use of this automated system, there is a constant search for the most recent vulnerability on the servers and systems, security patches and bug fixes. It can also prevent imminent attacks from malware and other intruders with any aim of damaging the network. It ensures these security patches are applied on time to avoid security breach, data loss and legal penalties (Florian, 2010).
- **Compliance:** Compliance is an essential key for companies to implement patch management solutions. Automating the system ensures best practices are met for security considerations. Listed below are some of the standards related to IT Security.
  - Family Educational Rights and Privacy Act (FERPA)
  - Federal Information Security Management (FISMA)
  - Health Insurance Portability and Accountability Act (HIPAA)
  - Payment Card Industry Data Security Standard (PCI DSS)
  - Sarbanes-Oxley Act (SOX)
  - Government Connect Secure Extranet – Code of Connection (GCSx CoCo)

  It ensures compliance with the above standards and many more thereby avoiding legal and financial fines or even loss of business in some cases.
- **New Features:** With the update on the patches, it ensures the new functions and features are added for extensibility support on additional platforms.
- **Productivity:** It results to a more protected system and increase in stability thereby speeding up the organizational process for the benefit of all employees as an increase in productivity can be easily measured from the output of the information technology (IT department.

A very good example of the application of an automated patch management process is when Staggs (2011) conducted a research on implementing an industrial automated patch management program (IACS) and addressed the issue of installing patches with no interruption in the manufacturing industry. For a smooth run of the automated process, different cross-functional teams were involved in the deployment of patches and the systems were designed in a way to support patch management. Automating a patching system does not result to taking the job from

the department responsible as it is still required of them to test the patches and certify them ready before they are applied on the network. However, it now reduces the amount of time required to test and analyze critical software and hardware patches as it makes it faster and more efficient. It also reduces the amount of mistakes that would be originally introduced into the system by the manual patching process. (Paranet, 2015) Some of the patch management tools which are currently in existence includes; Comodo Patch Management Software, Cloud Management Suite, Kaseya, SysAid and ManageEngine.

Conclusion

With the criticality of software patches constantly in the spotlight, patch management has an essential part of information security. This research was able to successfully highlight and describe the four fundamental phases in a patch management process; assessment phase, identification phase, evaluation and planning phase and the deployment phase which can be applied in any organization. Further breaking down the process into critical elements for better understanding in carrying out the roles and responsibilities. Automating the patch management process is very important for cost effectiveness and higher productivity. Andrew (2005) identified five Ps (Plan, Prioritize, Policy, Performance and Products) which are crucial for an effective patch management. Integrating these P's with patch management solutions would allow for automation of task distribution and applications. Planning is an integral part of patch management as it important for the company to know the potential vulnerabilities by risk assessing the business, taking stock of what it has and where it is located so as to be able to check for the vulnerability status as at when required. It is also paramount to prioritize the vulnerabilities by weighing the most crucial so as to prevent instabilities in the software programs. Developing a patching policy is also necessary so as to specify the order of the application of patches and the procedures for evaluating the severity of new alerts. Performance of each patch is required before the application for identification of risks and proper authentication. Choosing the right product solution would also save the companies the burden of constant patch deployment and solution.

References

Furnell, S. (2016) Vulnerability Management: Not a Patch on Where We Should Be? University of Plymouth. Elsevier. Network Security. Retrieved from: https://doi.org/10.1016/S1353-4858(16)30036-8

HCPro Inc (2018) Understanding the Challenges of Patch Management. ProQuest. Briefing on HIPAA. Retrieved from: http://search.proquest.com.jproxy.lib.ecu.edu/docview/2038582655?accountid=10639

Lavine, L. (2016) The Importance of Software Patch Management. Dental Products Report. Retrieved from: http://search.proquest.com.jproxy.lib.ecu.edu/docview/1818045391?accountid=10639

Andrew, C. (2005) The Five Ps of Patch Management: Is there a simple way for businesses to develop and deploy an advanced security patch management strategy? Elsevier. Computers & Security.

Gerace, T & Cavusoglu, H. (2009). The Critical Elements of the Patch Management Process. *Communications of the ACM – A Blind Person's Interaction with technology*. Retrieved from: doi: 10.1145/1536616.1536646.

Mathivanan, V. (2017) Tearing up the Traditional Approach to Patch Management with Automation. *Database and Network Journal Vol. 47*.

Palumbo, T. (2015) Patch Management: The Importance of Implementing Central Patch Management and Our Experiences Doing So. Proceedings of the 2015 ACM Annual Conference on SIGUCCS Pages 105 – 108. Retrieved from: 10.1145/2815546.2815561

Sihvonen, H.M., & Jantti, M. (2010). Improving Release and Patch Management Processes: An Empirical Case Study on Process Challenges. *IEEE Fifth International Conference on Software Engineering Advances*.

Staggs, K. (2011) Implementing an IACS Patch-Management Program. *Chemical Engineering Progress. American Institute of Chemical Engineers*. Retrieved from: http://search.proquest.com.jproxy.lib.ecu.edu/docview/870838008?accountid=10639

Mohammadi, H. (2014) A Systems Engineering Framework for Implementing a Security and Critical Patch Management Process in Diverse Environments. (Academic Departments' Workstation) The George Washington University. Retrieved from: http://search.proquest.com.jproxy.lib.ecu.edu/docview/1500834233?accountid=10639

ComodoOne (2017) Why Do You Need Automated Patch Management: Reasons Explained. Retrieved from: https://one.comodo.com/blog/patch-management/why-automated-patch-management.php

Pratt, M.K. (2018) 6 Steps for a Solid Patch Management Process. CSO. Retrieved from: https://www.csoonline.com/article/3268273/patch-management/6-steps-for-a-solid-patch-management-process.html

Cavusoglu, Cavusoglu, and Zhang. (2008) Security Patch Management: Share the Burden or Share the Damage? 658 Management Science 54(4), pp. 657–670. INFORMS, http://dblp.uni- trier.de/db/journals/mansci/mansci54.html#CavusogluCZ08

New Boundary (n.d.) Patch Management Best Practices. Windows Security. Retrieved from: http://www.windowsecurity.com/uplarticle/Patch_Management/ASG_Patch_Mgmt-Ch2-Best_Practices.pdf

CDC (n.d.) Vulnerability Management Life Cycle. National Program of Cancer Registries (NPCR). Centers for Disease Control and Prevention. Retrieved from: https://www.cdc.gov/cancer/npcr/tools/security/vmlc.htm

Redhat (2015) Life-cycle of a Security Vulnerability. Redhat. Retrieved from: https://access.redhat.com/blogs/766093/posts/1976453

Flexera (n.d.) Secunia Research turns indiscriminate information into verified intelligence. Retrieved from: https://www.flexera.com/products/software-vulnerability-management/lifecycle.html

Newman, L.H. (2017) Equifax Officially Has No Excuse. Wired. Retrieved from: https://www.wired.com/story/equifax-breach-no-excuse/

Fisher, T. (2018). What is a Patch? Lifewire. Retrieved from: https://www.lifewire.com/what-is-a-patch-2625960

Barreiro, A. (2011). Back to Basics: A Four Phase Approach to Patch Management. TechRepublic. Retrieved from: https://www.techrepublic.com/blog/it-security/back-to-basics-a-four-phase-approach-to-patch-management/

Tejaswi, S. (n.d.) Effective Patch Management. Vmoksha. Retrieved from: https://vmokshagroup.com/blog/effective-patch-management/

TAG (n.d.) Threat, Vulnerability, Risk – Commonly Mixed Up Terms. Threat Analysis Group. Retrieved from: https://www.threatanalysis.com/2010/05/03/threat-vulnerability-risk-commonly-mixed-up-terms/

Fisher, T. (2018) Patch Tuesday. Lifewire. Retrieved from: https://www.lifewire.com/patch-tuesday-2625783

Chapple, M. (2017). Automation, Process Improvements Simplify Patch Management. StateTech. Retrieved from: https://statetechmagazine.com/article/2017/01/automation-process-improvements-simplify-patch-management

Florian, C. (2010). 5 Benefits of Automating Patch Management. TechTalk. Retrieved from: https://techtalk.gfi.com/5-benefits-automating-patch-management/