Relationship of Configuration Management, Security Configuration Management and Patch Management

Lynn Ray
University of Maryland University College
United States
loye.ray@faculty.umuc.edu

ABSTACT: *Having different configuration management processes causes problems for organizations. The problem with different hardware and software configuration management processes is a lack of common methods and communication. A centralized configuration management process for handling all processes is necessary. Investigation of literature and common practices used in configuration management, security configuration management and patch management were examined. Also the author's experience was used to bring out experiences and practices used. This paper describes the relationship of the configuration management, security configuration management and patch management processes. There is a direct relationship of all three concepts in managing security risk in an organization. It emphasizes the building of a strong configuration control board process with leadership to overcome the problems of separate management processes. The result is an improved enforcement method to standardize changes that could negatively affect an organization.*

Keywords: Change Management, Configuration Control Board, CCB, Configuration Management, Patch Management, Security Control Management

Received: 1 September 2014, Revised 9 September 2014, Accepted 20 September 2014

1. Introduction

Today's businesses rely heavily on enterprise networks and systems to be available 24/7. They provide the backbone for the business by supporting web-based applications that are mission critical (Kerravala, 2004). These systems can come under attacks from malware and intrusions. Thus these computer and software systems require routine changes to combat attacks and improvements. Computer systems operating systems and software packages need patches to keep systems up to date and protected against these threats. According to Bellovin and Bush (2009) a configuration management process is critical to having safe and secure enterprise network.

However, today's configuration management processes can be inconsistent throughout an organization. These inconsistencies come about when individuals are allowed to perform actions without a central managed process. These can result into misconfigurations that can reduce performance and security protection (Bellovin & Bush, 2009). Sometimes software configurations may not be documented before being deployed. When they are documented it's usually after the configuration change and some information may have been lost. This can be from lapse of memory or personnel changes. Also others may not

be aware of any changes and the changes disrupt normal business processes or information technology services. An example can be allowing users to make their own updates without telling anyone until something fails (Muhammad & Sinnott, 2009). The reason for this is a lack of communication between effected parties within the organization.

One challenge with configuration management is the interactions of two or more configurations upon the system. Making the right configuration changes to individual system pieces may cause dangerous problems when they interact (Bellovin & Bush, 2009). For example: the combination of behaviors involving a correct Java configuration and firewall rule change can create a security weakness (Bellovin & Bush, 2009). Hackers could find a way of using Java to get through a firewall and into a protected network.

Another challenge occurs when organizations have different configuration management processes. These different processes may contradict each other and hence contribute to disruption of services. The problem with different hardware and software configuration management processes is a lack of common methods and communication. A change to one configuration area may require a change in another (Tomayko, 1990). This can cause a failure to maintain confidentiality, integrity and availability, and result in severe problems in an organization's operation, assets or personnel (FIPS Pub 199, 2004). The answer is a centralized configuration management program that can tie these processes together.

This paper is broken up into several sections describing how this is a problem. The first describes the relationship of the configuration management, security configuration management and patch management processes. The next section describes the building of a strong configuration control board process with leadership to overcome the problems of separate management processes. The result is an improved enforcement method to standardize changes that could negatively affect an organization.

2. Relationships

As it has been presented, there is a relationship among configuration management, security configuration management and patch management methods. One needs to understand this relationship to see the problems with separating each from the others.

2.1. Configuration Management

The basic configuration management process involves management of the hardware and software configurations within an organization. This involves maintaining and controlling a configuration baseline related to the last known user requirements in the current build (Johnson, Dempsey, Ross, Grupta & Bailey, 2011). A configuration management process includes the aspects of identification, control, status accounting, and audit and review (Dart, 2007).

Identification involves knowing the structure and components of the item being changed. Knowing how the system is constructed helps one understand the relationships of how the configuration change can affect the operation or security of the system. The control aspect is used throughout the system or software lifecycle. It involves placing controls when a release or update is being done against the established baseline. This baseline provides a means to ensure system structure or configuration has a point of reference to go by during the change. Status accounting involves collecting and maintaining information about the baseline and changes made to the system (Dart, 2007). Keeping this statistical information can help track progress and troubleshoot problems during changes. Lastly, auditing and checking the final result of the change can help ensure that the change was completed as planned. It also can ensure that the change process was consistent between other changes.

Changes to a configuration come from user requirements and may not be communicated outside the software or hardware development process. These are called change requests and should be tracked throughout the lifecycle of the system. The configuration management process uses change requests to create builds and help track changes to a system. It also provides for versioning to maintain baselines based on this process. A version number is associated automatically with changes and helps to define what is the current system configuration (Dart, 2007). When change requests are received and recorded, they provide a history of what was involved in the change. This history provides how the system evolved by showing what was changed and why (Dart, 2007). It includes specific components and configurations, describes the baseline and how changes have affected the baseline. This can be risky because the change can open the system to attack from hackers if the confidentiality, integrity and availability are not protected adequately. This deficiency can reduce efficiency and degrade the systems capability to support the organization as well as damage assets and help personnel (FIPS Pub 199, 2004).

The configuration management process involves a sequence of steps that defines what needs to be done with each change request. This process is a plan on how and who is involved with conducting each step (Dart, 2007). The configuration management process can be a manual or automated process that an organization follows (Dart, 2007). A manual process can be time consuming and error prone. This is especially true in large complex systems and networks (Chen, Al-Nashif, Qu & Hariri, 2007). Change requests in the manual method are written up and mailed or emailed to each member of the configuration control board. This can be inefficient and lead to errors. The automated method cuts down on the processing time and generates an automatic record for change requests. When a change request is filled out electronically and processing the same way, this cuts down on the time to process and approve the change request. Each member of the configuration control board can review and submit their question to the originator. Approvals can also be automated. It can also speed up revising the baseline configuration.

2.2. Security Configuration Management

Security configuration management is similar to configuration management in that security requirements are used to deploy systems to protect an organization. However, the requirements are based on risks and not user requirements. The aspect of proper configuration management is critical to deploying and maintaining software–based systems (Cavusoglu, Raghunathan & Cavusoglu, 2009). This is because there is a balance between the need for security protection and operational performance. The use of intrusion detection systems tends to help with this balance (Alsubhi, Alhazmi, Bouabdallah & Boutaba, 2012). Intrusion detection systems can defend against attacks while preserving network performance if configured correctly. However, there is a risk when using these and other security devices. The risk comes from the lack of proper configuration of security devices architecture. A good example is out of the box default configurations. Without proper security configuration management, these default settings can degrade the performance of the device and open networks to attacks. To help with tracking these is to use traceability to describe the risk requirements and how they were implemented (Mohan, Xu & Ramesh, 2008). A solution is to use traceability tools to link these items to facilitate security configuration management. Mohan, Xu and Ramesh (2008) found that traceability could be applied to software development practices, which are somewhat similar to security updates. Both software and security updates are continuous practices. Using a traceability tool can help track changes to systems and help troubleshoot problems. This can help achieve optimal performance of security devices and reduce the vulnerabilities and risk to computer networks.

Just like software versions, security deployments also track changes using versioning. This is done when updating such software packages as McAfee, Cisco, Microsoft and Juniper. Devices such as firewalls and intrusion prevention systems need frequent changes and updates. Using change management practices can help in keeping track of what has been updated (Capilla, Duenas & Krikhaar, 2012). This way security professionals can quickly determine if required security patches or updates have been deployed. However, there are multiple configuration changes being done manually. The result can be false alarms due to misconfigurations (Cavusoglu, Raghunathan & Cavusoglu, 2009). For firewall systems this is the configuring and maintaining firewall rule configurations that allow or deny network traffic through the firewall. Using either signature-based or anomaly-based intrusion detection, maintaining some record of the configurations is useful in ensuring effective performance from security devices. The deployment of configurations to one security device and impact other devices configurations (Cavusoglu, Raghunathan & Cavusoglu, 2009).

Security configuration management also uses a baseline like configuration management but pertains to security measures only. It provides a starting point where changes to the security measures begin (Joint Task Force, 2009). While configuration management doesn't, security configuration management maintains protection of the confidentiality, integrity and availability of an organizational system. Security configuration management usually has its own processes that are separate from the configuration management processes making it unaware of changes from other hardware and software systems being done under the configuration management processes. The risk is that security baseline measures may fail to maintain protection of current or new systems. Like configuration management, the security configuration management is approved by the organization who accepts the risks and that the approved security measures will protect the organization (Joint Task Force, 2009).

2.3. Patch Management

Patch management is a valuable part of configuration management and management of security posture of any computer or software system (Cavusoglu, Cavusoglu & Zhang, 2008). However, it can be very difficult to execute and costly to maintain. The results of not patching can lead to bigger costs in data breaches and company reputation. Therefore, security patches have been called the panacea for handling security related vulnerabilities (Cavusoglu, Cavusoglu & Zhang, 2008). Companies don't like to expend limited funds for doing this kind of preventive care for several reasons. One of the reasons is that there are a large number of vulnerabilities to fix and they occur frequently. For a large company with several computer system and software

vendors could see tens of upgrades a week. Each of these can be labor intensive and time consuming to implement across an enterprise (Cavusoglu, Cavusoglu & Zhang, 2008). However, one has to look at the consequences of not applying the patches. One can think of what are the losses to the business if a potential attack disrupts services or obtains personal information from customers (Muhammad & Sinnott, 2009). Also manual patch management systems can compound this problem by taking more time than an automated system.

To handle these it is recommended that vulnerabilities be grouped based on the rating of the vulnerability being resolved. High vulnerabilities require the highest priority over others. Next all upgrades and patches should be tested before deployment. This way one can determine if any patch may damage or cause problems with systems. Patches may fix some problems but cause other unexpected ones. Also patches come from different vendors and can't be deployed in a standard way (Cavusoglu, Cavusoglu & Zhang, 2008). Some patches may automatically be downloaded while others require a search on their vendor web site to find. Lastly, in some cases patches require one to reboot the system for the installation to be completed. Rebooting a server or restarting databases require system downtime that can impact service to customers.

Patch management is also similar to the other management systems. Patch management also uses a baseline of what patches have been applied to a hardware, software or security system. The administrators doing patch management perform their work usually outside of consultation from configuration management and security configuration management processes. They use such ideas as Microsoft Patch Tuesday for implementing patches to Microsoft products. These patches are usually reviewed for affects to current systems before installing them. To overcome this is to get executive support in of the security risks and how patch management can reduce threats and vulnerabilities. A supporting argument is that 95 percent of data breaches could have been avoided if one followed a dedicated patch management program with configuration management oversight (Grace & Cavusoglu, 2009). Once these senior c-level executives realize the impact to security risk, they can see the benefits of the program.

3. Working Together

The challenge now is how to make these different controls work together and resolve any differences.

3.1. Use of a Configuration Control Board

Overseeing of the configuration management process should be to ensure that all policies and procedures for making changes to any configuration be followed (Dart, 2007). To do this, a manager creates mechanisms for submitting change requests, managing reviews by others and getting approvals. This process involves using a change control board. The configuration control board provides a means to approving change requests before they are implemented.

A configuration control board comprised of individuals from different areas that support hardware, software, security, and information technology support will help make individual processes work better. This is because each can communicate pending changes to systems that could affect the baseline configurations, security or operation of an information system (Tomayko, 1990). The configuration control board should utilize a standard form and process that allows each area to communicate these changes (Dart, 1991). The form will contain information about the change and risks to other systems. A process would be devised and used to route the form to all configuration control board members. All changes to any configuration or operation of an information system must be approved to avoid degrading or failure of mission systems (Milligan & Bellagio, 2005; Tomayko, 1990). These processes will be well defined and reviewed annually for improvements. The board may meet in person but could be done via automated means such as email or centralize document systems such as SharePoint. Routing the form gives all parties time to review and pose questions to ensure any impacts to systems have been adequately addressed. The configuration control board should be lead by an Information Technology Director or other senior manager to ensure the process is followed. These individuals need to have the authority to approve changes to the baseline while understanding the risks to the organization (Johnson, Dempsey, Ross, Grupta & Bailey, 2011; Tomayko, 1990). This way the organization can approve the changes and the risks they may pose to baseline configurations. Still the process and configuration control board can establish and approve baselines for hardware, software and security configurations. The use of version control for each of these can help communicate where systems are and resolve problems when changes are reviewed for consideration (Estublier, 2000). This can be lacking in separate processes where one can lose control of what version is being used or how changes may have affected it. The configuration control board should be able to audit, track and control versions of each baseline to reduce risks (Milligan & Bellagio, 2005). They can determine who is allowed to make changes and what changes are permitted.

3.2. Establishing an Automated Patch Management Program

Setting up a patch management program requires some simple steps be followed. There should be dedicated resources of personnel made available to run the program. Part-time resources don't always have the dedication needed to make the program successful. Another step is to conduct and maintain a thorough inventory of all technology that could need patching (Grace & Cavusoglu, 2009; Meyer & Lambert, 2007). As systems are added or retired, the list needs to be updated according. This inventory will help in determining what needs patching and where to find the devices. Next a process needs to be created to identify where vulnerabilities are and what patches are needed (Grace & Cavusoglu, 2009). Using the inventory can best monitor what needs patching. Also using automatic vulnerability scanning tools can help in finding weaknesses and where patches are needed. Before deploying any patch, one needs to test it in a lab or testing environment (Meyer & Lambert, 2007). This way any negative effects can be found before they can affect operations and services to customers. This can save time and money in resolving problems when patches cause problems. Lastly, one needs to monitor the system performance a few days after a patch deployment. This way one can determine if there were any problems missed in the previous testing. All these help to establish a baseline of how systems are protected.

One should also setup metrics to measure how well their patch management program is working. These are used by management to better track the process and find weaknesses in the program (Meyer & Lambert, 2007). The effectiveness can be measured by looking at how many systems have been patched at any given time (Grace & Cavusoglu, 2009). While it may not achieve 100% it should be in the high 90-95% range. Setting a goal in this range is challenging but achievable. Another method is to determine the susceptibility of systems being attacked using the number of vulnerabilities found and the number of systems patches (Meyer & Lambert, 2007). This method may be more effective a measure than the earlier one. Along with the baseline, metrics can be a powerful combination in an automated patch management program.

3.3. Communicating Changes

Building or utilizing an automated change request process can speed up communication and approvals. Using easy to fill out forms provides a way of describing the change and why it's needed. It also documents the effected systems and risk that the change can impact. A simple fill in the blanks and check boxes helps speed up the process. The change request form can be stored in a configuration management system that can be accessed by others. These computer forms can be used with workflows to route the change request among members of the configuration control board. The workflow would send an email message to each member that a new change request is available for his or her review and approval. It becomes a centralize storage of any change request that could be accessed at anytime. This can be important when troubleshooting problems and updating baselines.

Another method of getting the different processes to work includes involving security in each phase of the development life cycle. This way security can communicate how the impact of changes could cause on the operations and baseline security configurations (Milligan & Bellagio, 2005). Also patches to software need to be communicated before they are implemented so as to warn others of possible effects to system operations. The configuration control board can be a vehicle for accomplishing this. One should also do testing and publish results to the configuration control board members to any findings that may affect current operations. Lastly, the configuration control board will maintain a record of all meetings and the results for future review (Johnson, Dempsey, Ross, Grupta & Bailey, 2011). These improvements in communications can support working relationships among individuals involved with configuration control, security configuration control and patch management.

3.4 Benefits

Creating an effective configuration management program involving patch management and hardware/software systems can be a critical impact to an organization (Kerravala, 2004). Systems can provide improved efficiency and productivity while reducing the weaknesses to attackers. It can also make configuration changes faster and more accurate than manual methods. Using automated configuration management processes for patch management and change request coordination can improve tracking of changes and auditing (Kerravala, 2004). This helps keep the baseline configurations up to date to better troubleshoot problems.

Therefore, bringing each of these processes into a central controlling process like the configuration control board can reduce the negative effects of keeping them as separate identities. The configuration control board can ensure that the confidentiality, integrity and availability of each information system are adequately protected against risk while meeting user requirements. Also regular processes like patch management can still be performed but with consideration for the impact the changes may cause if not communicated properly. Lastly, establishing a baseline for configuration management, security configuration management

[15] Milligan, T. J., Bellagio, D. E. (2005). What is software configuration management?, http://www.ibmpressbooks.com/articles/article.asp?p=390813

[16] Mohan, K., Xu, P., Ramesh, B. (2008). Improving the change-management process. *Communications of the ACM* 51 (5) 59-64.

[17] Muhammad, J., Sinnott, R. O. (2009). Policy-driven Patch Management for Distributed Environments. *In: Proc. of the Third International Conference on Network and System Security (NSS '09)*, pages 158-163. IEEE Computer Society, Oct. 2009.

[18] Tomayko, J. E. (1990). Software configuration management, http://www.sei.cmu.edu/reports/87cm004.pdf