Global Information Assurance Certification Paper

Designing and Building an Effective Security Alerting Process
Nina Ferguson GSEC Practical Assignment, Option 1, Version 1.4b

Abstract

In a perfect world, system or product vulnerabilities would be mitigated or eliminated by the timely installation of security patches or upgrades. The goal is to deploy solutions or patch systems in advance of any potential threats or exploits. Deciding which systems to patch and when to patch them can be a daunting task to a system administrator. The mind-boggling number of security alerts that are issued weekly presents a formidable challenge in deciding which alerts require immediate attention. These activities are all time-consuming, but they can be organized using a system management approach.

Managing change in the environment requires a well-ordered process that is documented, understandable, and measurable. One important facet in managing change is the recognition of a problem and the notification of impacted parties to resolve the issue. This paper will discuss the design and creation of technical and process-oriented solutions to reduce the risk of security vulnerabilities and malware (i.e., malicious code) in the environment. This document will help the reader understand how to build a security alerting (notification) system as a pre-requisite for patch control in a large, distributed corporate environment.

Understanding the Security Threats

We live in dangerous times. The constant threat of network exploits, viruses, worms and Trojans requires that we are vigilant in our security protection strategies. In 1995, CERT reported 2,412 security incidents. Last year, the number of incidents reached 52,658. [1] In just the first quarter of 2003, the number of incidents reported was 42,586. [2]

One critical security problem is the threat of viruses and the impact to the client community when an attack occurs. Advance notification of an impending malware attack is crucial to preventing a full-scale network outage. The impact of recent virus events like Funlove and Klez may be reduced if preventative measures are taken to install current anti-virus signatures or attachment blocking at the email gateways, internal email servers and in the desktop/file server environments.

Another threat vector, which has received widespread media attention, is Internet-based attacks which take advantage of system vulnerabilities. The recent Slammer worm caused denial of service problems for compromised, internal hosts. This vulnerability could have been mitigated by the timely

installation of a Microsoft patch for SQL servers. This patch was available for installation several months before the attack occurred. An alert notification and patch control process could have prevented this attack.

Lastly, security flaws in third-party products like Oracle and Sendmail present additional problems, where there may be business application dependencies which impact one's ability to perform software upgrades. Patches must be tested before implementation to reduce the risk of software incompatibility between applications and the new software patches. Conducting application testing may delay the deployment of necessary patches and certainly impacts a company's capacity to respond to emergency software changes.

Exploits are written to take advantage of known security flaws. Malware writers understand that the inertia associated with patch management contributes to their ability to succeed. Security professionals are constantly challenged with convincing their IT counterparts to respond to security alerts in an appropriate timeframe. The prevailing opinion among system administrators and developers is "if it ain't broke, don't fix it." Unfortunately, that consensus will eventually come back and haunt the system owner as intruders are becoming more creative in their attempts to compromise systems.

Background and History

Historically, the management of the computing infrastructure has been assigned to the Information Technologies organization; however, in a large, distributed computing environment, there may be a variety of O/S platforms and business units that are responsible for those resources. In addition, standard processes to manage the environment may not have been adopted. When this scenario occurs, it is difficult to establish inventory and lines of responsibility and accountability. Attempts at vulnerability remediation will meet with varying degrees of success. One school of thought recommends that system administrators lock down servers and remove as many threat vectors as possible from the outset. The idea is to anticipate the most common types of vulnerabilities and take away those avenues into the network before an attacker finds them. [3] This methodology works well in a closely, administered environment, such as a firewall system. However, on many internal servers, hardening systems for security may not be an option. In a climate where corporate downsizing is a reality and resources are limited, system administrators have enabled services to allow ease of remote administration. Operational processes have been built to take advantage of the full range of the computing resource. Business developers, in their haste to bring products to market, have coded functionality that use known flaws in the operating system to expedite the delivery of their software products. Increasingly, Internet access to internal applications is a business requirement, and, thus, hardening systems prevents the free exchange of data between the applications and the clients they serve.

In the past, to mitigate the risk of vulnerable systems, a company might have chosen to incorporate host vulnerability assessment tools in the environment. It is my experience that these tools must be carefully selected. Some of the commercial tools generate a mountain of reports that no one will read; moreover, there may be a high incident of false positives which damages the credibility of those tools. If host vulnerability tools are used, I would suggest that an assessment of the environment be made and certain suspect, false positive conditions be eliminated from the vulnerability scans.

Many companies use a more traditional approach. To combat malware, they may use a multi-faceted protection scheme that involves deploying anti-virus signatures on a set schedule to minimize the impact to users and the support organizations. Software changes can also be managed by deploying service packs or security bundles from the vendor. I have found that the software quality with service packs and patch bundles is better than the quality of individual patches or hot fixes. However, there is a trade-off in system security. Typically, a company may not be positioned to respond quickly to security incidents.

The CodeRed and Nimda worm events dramatically emphasized corporate America's lack of preparedness for Internet-based attacks. The industry, as a whole, was not aware of the vulnerabilities nor did they know what systems would be impacted by the attacks. Many companies were scrambling to figure out how to react to these events. Hundreds of hours were spent patching IIS servers. In some cases, companies disconnected the compromised systems from their network because they could not locate the system owner. The identification of server owners is only one key factor in helping to reduce the risk of malware and vulnerabilities in the environment, but the root problem of vulnerability awareness and notification must still be addressed.

A good vulnerability awareness and notification system is necessary to inform impacted system owners that potential flaws exist on their systems and to provide information on how and where to get assistance in resolving these system vulnerabilities. However, just notifying clients of new vulnerabilities does little to remediate the vulnerability. An organization still has the task of implementing the necessary patches or software upgrades to resolve the vulnerabilities.

This point was highlighted by the recent Slammer worm incident earlier this year. Huge traffic performance problems occurred as the number of infected servers increased and traffic volumes rose. Again, companies were faced with the dilemma of patching thousands off Windows servers in a very short amount of time. A reactive approach to patch management does not prevent nor does it reduce the risk of vulnerabilities and malware in the environment. This patching philosophy is out of sync with the current security climate. Proactive solutions are needed to address evolving security requirements. This requirement can

best be met with a deployment of a process that incorporates rapid identification, notification, and response to security vulnerabilities.

Setting the Framework for the Process

Security alerting is a complicated process involving many facets. Companies spend more than $2 billion annually on patch research and deployment, according to Aberdeen Group Inc. in Boston.[4] Simply applying all patches when they are announced by the vendor(s) may not be the best approach. However, waiting until the next patch cycle is equally objectionable, since exploits may be readily available to take advantage of these vulnerabilities

When embarking on building a new or revising an old process, it is wise to obtain senior management support before beginning, particularly if it involves changing the software deployment philosophy. System owners are loathe to patch their servers if it impacts the availability of their systems and, specifically, if the patch requires a system reboot. Another issue is patch quality. System administrators are quick to point out that patches are often poorly written and may not even work as advertised. There is a fear that the software patch may break the very application it was intended to repair. The notion of security versus availability is a management dilemma. Compounding this problem is the existence of service level agreements which dictate application availability and impact the system owner's ability to make changes to the system, particularly if an outage is involved.

They key to implementing an effective security alerting system is to begin with a good inventory. If an inventory is not available, it would be extremely difficult to determine what needs protection. The inventory should specify: [5]

- The systems that make up the environment
- Their operating systems and applications, including version number
- What patches have been applied
- Ownership and contact information (important to large and far-flung companies)
- Any known but unpatched threats to these systems and vulnerabilities in them

I would also recommend taking a baseline the environment. Baseline is the process of bringing the computers in an environment to a standard software baseline – that is, with the same software versions and software updates. Baselining includes the following steps: [6]

1. Generate an inventory of hardware (see above).
2. Use the information obtained from the inventory to define standard software baselines for all computers.

3. Perform an audit to determine which computers meet their baseline and which do not.
4. Take the necessary actions to bring the non-compliant computers up to their required baseline. This involves installing service packs and other software updates or even upgrading software versions.
5. Audit the environment to ensure the standard software baselines are met.

In a patch management system, one suggested approach by security software vendors is to:

- Develop a patch network
- Buy time by prioritizing
- Evaluate before you patch. [7]

Using these guidelines, I would recommend the following key elements in your process:

- Review security alerts from reputable security vendor websites:

Review alerts from several security alerting vendors as well as communication from software vendors. Read news publications on the Internet to help identify new vulnerabilities. Determine whether the alert applies in your environment.

- Post the alert in a security alerts database:

You may want to build a database or keep track of the alerts using a spreadsheet. Post alerts for viruses, worms, Trojans and all relevant security vulnerabilities for operating systems and products. Each posted alert should contain its criticality rating which reflects the urgency of the alert. Using the vendor's criticality rating may be a good option, but understand how their rating system is derived. There is a tendency for security alerting services to rate vulnerabilities on the high side.

- Notify impacted system owners via email:

Once posted, email the alert to all impacted or interested parties. Using the information compiled from the inventory, send notification of the alert to your contacts.

- Monitor receipt of response(s) from system owners:

System owners respond to each alert by providing acknowledgement of their impact to posted alert and a count of impacted systems.

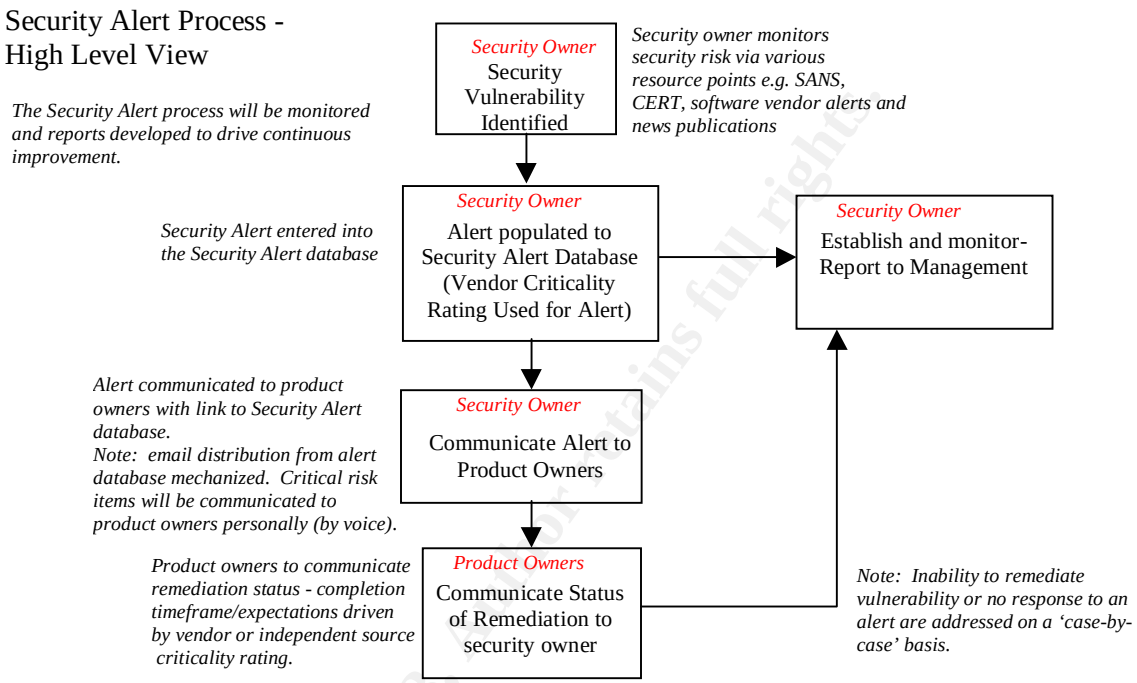
- Schedule patches or other remediation efforts in the environment:

Patches are scheduled for deployment based upon the severity of the alert. Using one industry standard, [8] remediate vulnerabilities using the following criteria:

- Critical alerts are remediated within 48 hours
- High level alerts are remediated within 5 business days
- Medium level alerts are remediated within 15 business days

- Low level alerts are remediated by the next patch cycle or 90 days

The process flow that is built is depicted in the following flow diagram:



Once the overall process is defined, meet with internal clients to discuss roles and responsibilities. To expedite delivery and monitoring of patch deployments, consider implementing automated software management tools that are readily available from various software vendors. The introduction of those tools will enhance your ability to respond to emergency events and to determine the status of your patch efforts.

Building a Security Alerting Database

At the heart of a security alerting system is the security alerting database. This database stores all of your security alerts, tracks progress towards remediating the vulnerabilities, and identifies all of your clients, the products and operating systems that are in use in your company.

The key functional tasks for your security alerting system are:

- Asset identification
- Record security alerts

- Notification process
- Monitoring process
- Reporting process

In constructing a security alerts database, several design considerations need to be addressed:

- What alerting sources can you trust? It is my experience that some sources did not provide enough information (e.g., no actionable activity for remediation is recommended). These types of alerts are simply advisories for the security owner to monitor and do not require action from the support organizations. I would recommend that you use sources that have a broad client base and are recognized leaders in security notification.

  In the market, there are a variety of sources that publish security alerts. While some alerting services are free (e.g., CERT, SANS), other alerting and threat assessment services are costly and the data is, typically, stored at the vendor's location. When purchasing a security alerting service, care should be taken to select an alerting service that allows you to choose what kind of alerts you want to see and to customize the type of reports that you receive.

- What resources did you want to protect? You may want to report security alerts for all of the O/S platforms in your environment, major network elements, and key products (e.g., databases, utilities, development tools).

- How do you identify your clients? Perhaps the most difficult task is to identify your client base and what they manage. Most of that information should be available in your asset inventory, but there may be gaps. Consider enlisting the help of key business leaders who can assist you by assigning security representatives to work with you to identify system owners.

- How does the client provide feedback for each alert? One solution may be to provide a web interface that allows the clients to update their status by alert. I recommend restricting access to the application via logon and an access control list. Clients should only view and update data for their own organization.

- What assessment rating (alert severity rating) should you use? Many of the alerting services have their own vulnerability assessment rating system and the criteria they use to rate an alert. I would recommend that you use the vendor's rating on the alert (if available) or the SANS (MITRE) CVA Priority rating system. In practical use, I have found that there is a

need for more research in vulnerability assessment to promote consistency and standardization of terms. Since the alert rating drives the response mechanism, I believe that a comprehensive industry standard should be adopted and used by all of the alerting services.

In addition to assigning a severity rating to the alert, you might also consider designating an exposure rating to each alert. The exposure level measures the risk in the environment and assesses how many systems are impacted by the alert, how easily the vulnerability can be exploited, and whether there is a current exploit of that vulnerability available. The exposure rating helps you to prioritize the alerts when multiple alerts are posted and gives your senior management a sense of how vulnerable your environment is to the security problem. The combination of vendor rating and exposure rating drives the response from your clients.

This figure shows an example of the exposure values and their associated rating criteria:

Exposure Attribute	HIGH	Medium	Low
Server or client compromise (number of impacted systems)	>50%	10-50%	<10%
Problem found in default configuration/installations	YES	YES	NO
Affected assets high value/business critical applications or core infrastructure	HIGH	MEDIUM	LOW
Network Infrastructure infected (DNS, routers, firewalls)	YES	YES	NO
Exploit code publicly available	YES	YES	NO
Technical vulnerability details available	YES	YES	NO
Difficulty to exploit vulnerability	EASY	MEDIUM	DIFFICULT
Attacker needs to lure victims to hostile server	NO	NO	YES

- What kind of reports do you want? Reporting metrics for compliance is a on-going activity. These reports are shared with senior management for both the security and client organizations. Compliance reports help identify problem areas in the organization and provide a measurement on the state of security within your company.

Establishing a Response Model

Once the alert is distributed, there are built-in expectations for response from your clients. Each alert is monitored weekly for client compliance and completion of work effort. The severity of the alert, as defined by the vendor, and the exposure rating, dictates the client's timeframe for response.

As previously noted, an aggressive remediation schedule (see reference note 8) is necessary to address the range of security alerts based upon criticality rating. In a large, corporate environment with many different technologies and high availability requirements, it is extremely difficult to meet these targets. An examination of these response timeframes shows that liberal amounts of time for testing, troubleshooting, and installation of patches is not built into the response schedule. The response model may be adjusted by assigning security zones with immediate protection provided for highly vulnerable systems (i.e., typically those systems that are outside of the Firewall). Internet-facing systems are patched within the response timeframes. A less aggressive patch schedule for internal servers and desktops (usually within 3 weeks) may be permitted, but consider prioritizing which internal servers receive the patch(es) first depending upon the alert that is posted.

In a critical alert situation, convene a SWAT (escalated response) to discuss the roles and responsibilities and logistics for resolving the vulnerability. As a short-term response, consider disabling the service that has the vulnerability. If a system becomes infected or compromised, it is removed from the network and steps are taken to patch or disinfect the impacted machine before it is brought online again.

It is a challenge to patch the systems in a timely manner so that you are not vulnerable to exploits that may be in the wild. It would be interesting to apply a risk management approach to the response model. Risk models based solely upon threat likelihood are hard to justify to management, but a model that establishes business-oriented security priorities may make more sense. [9] The combination of asset value, asset location, and threat likelihood are factors that should be considered when building a risk-based model. I believe additional research is necessary to further refine the alert response model and to mature the process.

Developing Vendor Alliances

Early in the development of your process, it would be a wise to include your major vendors in the identification, discussion, and resolution of new security alerts. Conduct weekly security calls with your major vendors to discuss the current week's alerts. Holding these meetings with the vendors also gives your

clients a chance to voice their concerns on the impact of the alert in their unique environments.

Patch quality in an accelerated patch deployment program is an important requirement. In my experience, I have found that vendor software patches are often buggy, particularly if the patch is rushed to the public without the proper internal vendor testing. This situation is exacerbated when exploits are made public and the patch has to be released before it is ready.

One approach to decrease the risk of installing faulty patches is to participate in a pre-release security patch validation program with one of your major vendors. Advance notice of an impending patch allows you to test the patch in a controlled test environment and discover any installation caveats.

Establishing vendor relationships, as a key strategy, is a good business decision. Ownership of the security alerts process is shared among all impacted parties – the security organization, your clients, and the vendor(s) who generate solutions for the alerts. It also holds the vendors accountable for the software patches they release and the solutions they recommend. Without the vendor's participation, they cannot begin to understand the amount of disruption and chaos that each vulnerability brings to the environment. There is value in partnering with your vendor(s) to resolve any issues with your patch management plan.

Implementation Strategies

Many roadblocks and issues will arise during the course of implementing a security alerting process. Undertaking the building of this process will generate many heated discussions and create resistance among your peers in the support organizations. This project is a huge effort to undertake and could very easily affect the working relationship and credibility you have with your clients. Several strategies I would recommend for a successful deployment would include:

- Establish clear objectives, define client requirements, and document the plan.
- Understand the environment you want to protect and your client base.
- Enlist senior management support for your project and communicate the plan and its objectives to them.
- Involve your clients in the planning and development of the process. Listen to their feedback on changes to the system. They can provide valuable information that will improve the process flow.
- Involve your vendors in the planning, development, and deployment of the process.
- Be very clear about the roles and responsibilities of each impacted party.

- Research commercial security alerting services. If time is scarce and your budget permits, you may want to invest in a security alerting service rather than build your own system.
- If budget is a concern, take advantage of free security alerts. For example, each week the non-profit, vendor-neutral SANS offers personalized "security Alert Consensus" reports that "summarize the vulnerability traffic of several major security mailing lists, broken down by software vendor categories. Additionally, if your network relies on a specific OS or vendor, make sure to subscribe yourself (or your team) to that vendor's announcement lists." [10]
- When a roadblock that occurs that you cannot resolve, leverage the experience and knowledge of your management chain. They may be able to resolve your problem, particularly if organizational politics are involved.
- Adopt a system management philosophy. It provides a roadmap for you on how to define a problem and identify the steps to take to resolve the issues that arise. By implementing a comprehensive process and the right combination of tools in this project, you will improve the security in the enterprise.

Roadblocks are a certainty with any large project. Overcoming these obstacles can be another matter. I would recommend avoiding the following pitfalls:

- Do not attempt to implement this process without the buy-in of upper management support. Your clients will offer resistance. They may even dismiss your project if you dictate the rules and expect them to follow.
- Do not underestimate the complexity and magnitude of the data gathering effort. If a data asset inventory already exists, leverage that information to begin building your security alerting database.
- Do not offer to resolve your client's problems. If you do, you own their problems and they become victims. Be very clear in defining the roles and responsibilities.
- Do not attempt to solve 'world hunger' with your process. Keep your goals and objectives within reason. Your scope should extend to only those areas you can control. Set milestones for your project and monitor your progress towards completion.

Benefits of a Security Alerting System

An organization that implements a security alerting system will realize several significant benefits. An immediate benefit is that there is more awareness of the vulnerabilities in the environment and better processes in place to quickly resolve security incidents should they occur. Are you immune to future security events like CodeRed and Slammer? Probably not. However, you are better positioned to isolate and resolve problems with certain systems that are vulnerable to the exploits and you can manage that effort without the struggles that were encountered in the past.

One pleasant outcome of implementing this process is the improved relationships you will enjoy with your clients and vendors. A trust relationship will be forged that did not exist in the past. The clients will have better software management processes and know their roles and responsibilities with respect to security. Your vendors will be more willing to help you resolve patch problems and identify new tools that will expedite the delivery and monitoring of security patches. The impact of this process can be measured in terms of goodwill generated among your peers, management, and the vendor community

You will also realize a reduction of risk in the environment. The average time to deploy software changes will significantly be reduced. The managed, response time approach to software changes will help you reduce the number of vulnerable systems. You may experience fewer virus incidents and be prepared to respond to network attacks by being able to focus on those systems that have not been patched.

The challenge of building a security alerting system can be daunting. With the proper planning, designing, and client/vendor participation, it can be a rewarding experience. I believe the benefits that an organization will derive from implementing this system will prove to be a worthwhile investment in time, energy, and resources.

References:

1. Fisher, Dennis. “Living with Worms, Viruses, and Daily Security.” eWeek. February 11, 2002. URL: http://www.eweek.com/article2/0,3959,40289,00.asp (June 20, 2003)

2. Shipley, Greg. “Are you Vulnerable?.” Network Computing. June 26, 2003. URL: http://img.cmpnet.com/nc/1412/graphics/1412f1_file.pdf (June 27, 2003)

3. Fisher, Dennis. “Patch as Patch Can.” eWeek. December 9, 2002. URL: http://www.eweek.com/article2/0,3959,758258,00.asp (June 20, 2003)

4. Collett, Stacy. “Manage those Patches!” Computerworld. July 15, 2002. (2002): 28

5. UlFelder, Steve. “Practical Patch Management.” NetworkWorldFusion. October 21, 2002. URL: http://www.nwfusion.com/supp/security2/patch.html (June 22, 2003)

6 “The Microsoft Guide to Security Patch Management.” July 3, 2003. URL: http://microsoft.com/downloads/details.aspx?FamilyId=73AC38B7-5826-421D-99E8-CDCC608B8992&displaylang=en (July 8, 2003)

7. Collett, Stacy. “Manage those Patches!” Computerworld. July 15, 2002. (2002): 28

8. “About the CVA Process and CVA Priority Ratings.” URL: http://www.sans.org/newsletters/cva/ (June 29, 2003)

9. Barwise, Mike. “Calculating the Risk Equation.” Computer Weekly. January 16, 2003. (2003): 30

10. Stafford, Jan. “Dos & Don’ts: Smoothing Out Patch-Management Woes.” SearchEnterpriseLinux.com. April 2, 2003. URL: http://searchenterpriselinux.techtarget.com/originalContent/0,289142,sid39_gci891798,00 (June 30, 2003)