# Global Information Assurance Certification Paper

# *Vulnerability Risk Reduction and Baseline Configurations: The Effective Way*

**Heath McGinnis, CISSP**
**GSEC Certification, Version 1.4c Option1**

**Submitted**
**20 December 2004**

**Table of Contents**

## Abstract

This paper will address the implementation of an effective electronic Vulnerability Reduction Program within an organization. We will discuss those critical areas on which vulnerability reduction processes are dependant in order to be successful and must be in place in order to affect an efficient Vulnerability Reduction Program.

## Introduction

With the advent of recent US Laws regarding gross negligence, organizations today know that electronic risk reduction is imperative to maintaining a functional business infrastructure. The problem that most of these organizations face is how to implement an electronic risk reduction program with the limited resources that they possess. These resources are not limited to the tools needed to conduct this effort but include the personnel and knowledge needed as well. Knowing that Risk is an equation, represented below, organizations make one logical conclusion. Bring any of the variables to zero and the risk will net at zero. According to Peter Tippett of the TruSecure Corporation the following is a representation of the Risk Equation:

Risk = Threat * Vulnerability * Cost

*Threat is the frequency of potentially adverse events.*
*Vulnerability is the likelihood of success of a particular threat category against a particular organization.*
*Event Cost is the total cost of impact of a particular threat experienced by a vulnerable target. (Tippett)*

Organizations are aware that they cannot directly control threats and that it is infeasible to operate an organization on systems that are not cost beneficial. With these realizations, organizations with limited resources take one of two actions; they either throw their hands up in the air because of the lack of resources/knowledge base or they attempt to address vulnerabilities directly with little more plan than to brute force their way through the process utilizing procured or downloaded tools. In the second case organizations become extremely frustrated because of the sheer volume of vulnerabilities identified or because of the inability to validate and prioritize the resolution of these vulnerabilities. While it is preferred to have a highly experienced security professional put a Vulnerability Reduction Program in place, that knowledge base is not required to begin the process. In the immortal words of John F. Kennedy, “The time to repair the roof is when the sun is shining.” A proper Vulnerability Reduction Program will encompass more than just vulnerability

identification. This program would include a Baseline Process, the Assessment Process, an Ongoing Maintenance Process, and the Vigilance Process.

## Baseline Process

While many believe that a vulnerability assessment program should revolve around the assessment process, the beginning and end of a good vulnerability assessment program revolves around the Baseline Process. All principle security practices are dependant upon one thing and that is the knowledge of what it is that is being protected. This concept can be referred to by a principal axiom put forth by Socrates, "Know Thyself". Meaning an organization must know what systems it needs to protect and how it needs to protect them. This being said, the Baseline Process of a Vulnerability Reduction Program is made up of two initial phases; the first being the Inventory Process and the second being the Hardening Process.

### *Inventory Process*

The Inventory Process allows an organization to identify or classify systems into various groups for easier handling. These groups can be broken down into any method that is appropriate for that organization. This grouping or classification is typically seen as part of a Business Impact Analysis and can be utilized in the process of identifying assets to be part of a Baseline. This is needed because not all assets will require the same security controls and responsiveness based on their criticality or function. For example (Table 1), it would not be appropriate to apply Windows 2000 Desktop security controls to a Windows 2000 Server. The key is to group likes so that systems can be addressed as a whole rather than on an individual basis. This provides a starting point for the baseline process prior to defining actual hardening controls.

| | Critical | Non-Critical |
|---|---|---|
| **Server** | | |
| HPUX | CH | NH |
| Win2k | CM | NM |
| Solaris | CS | NS |
| **Infrastructure** | | |
| Cisco | CC | NC |
| Nortel | CN | NN |
| **Desktop** | | |
| WinXP | X | NX |
| Win2K | X | NW |

*Table 1*

### *Hardening Process*

They key to the Baseline Process and the entire Vulnerability Reduction Program is in the Hardening Process. By defining, utilizing, and enforcing good Hardening Guidelines it is possible to address 80-90% of known vulnerabilities. (Muiccio) For this reason it is important that these Hardening Guidelines be well defined and tested. This being said it is imperative that organizations, especially those without extensive security experience, obtain tried and true Hardening

Guidelines from external entities as a good starting point. Good sources for Hardening Guidelines are www.NSA.gov, www.NIST.gov, www.CISecurity.org, and SANS.org. CERT provides the following basic guidelines within their Security Improvements Modules regarding the hardening of a device:

- *Determine the functions that you intend to support with your network server.*
- *If there are alternative ways of providing the same function, select the more secure way.*
- *Once you determine the minimal set of services and applications, ensure that only those are installed on the host.*
- *Eliminate any unnecessary open network ports.*
- *After you make all configuration choices, create and record cryptographic checksums or other integrity-checking baseline information for your critical system software and its configuration.* (CERT Security Improvements Modules)

The end goal of these steps is to put the device into a default deny posture, one of the most important security principles as discussed by Jonathan Feldman in his Network Magazine (Feldman). Meaning that the system will only support the minimum of services needed to support its users. Good Hardening Guidelines will account for this by defining what the minimum service requirements are for functionality and listing all applicable patches required to bring the system into a functional hardened state. Utilizing NIST's ICAT Vulnerability Statistics (Table 2) below we can see that over 90% of vulnerabilities are introduced either by the Operating System or by services running on the system itself.

| Exposed Component | 2004 | 2003 | 2002 | 2001 |
|---|---|---|---|---|
| Operating System | 124 (15%) | 163 (16%) | 213 (16%) | 248 (16%) |
| Network Protocol Stack | 6 (1%) | 6 (1%) | 18 (1%) | 8 (1%) |
| Non-Server Application | 364 (45%) | 384 (38%) | 267 (20%) | 309 (21%) |
| Server Application | 324 (40%) | 440 (44%) | 771 (59%) | 886 (59%) |
| Hardware | 14 (2%) | 27 (3%) | 54 (4%) | 43 (3%) |
| Communication Protocol | 28 (3%) | 22 (2%) | 2 (0%) | 9 (1%) |
| Encryption Module | 4 (0%) | 5 (0%) | 0 (0%) | 6 (0%) |
| Other | 5 (1%) | 16 (2%) | 27 (2%) | 5 (0%) |

***Table 2*** (NIST ICAT Vulnerability Statistics)

It is therefore concluded that it is more effective to define how systems should look in a hardened state prior to being implemented into the organization rather than chasing the potential thousands of vulnerabilities throughout the organization. This is the most import step to accomplish prior to conducting the first Vulnerability Assessment. Utilizing CERT's Vulnerability Statistics (Table 3), if we were to scan a device for all of the vulnerabilities identified from 2000-2004(Q3) for one device we would need to scan 14,123 vulnerabilities. Presuming that a server out of the box would only be found vulnerable to .5% of these vulnerabilities, this server would still require approximately 71 different actions to be taken on that device. If these vulnerabilities were addressed on this device as part of a standard build process or checklist (Hardening Guidelines) the efficiency of the Assessment Process would be greatly increased as we will see below.

***Table 3 (CERT Vulnerability Statistics)***

## Assessment Process

After defining appropriate Baselines the next logical step is to define the Assessment Process. The Assessment process is the heart of the electronic Vulnerability Reduction Program. In order to adequately define a functional Assessment Process we must cover first the typical problems related with vulnerability assessment programs, secondly the steps to executing an effective vulnerability assessment, and finally the expected outcomes or results from the vulnerability assessment process.

### *Typical Assessment Problems*

Traditionally the problems with vulnerability assessments are the lack of expertise and the lack of understanding by administrators/management in the true function of a vulnerability assessment process. The problem with expertise is that it is directly related to experience. Unfortunately in the understanding and implementation of a vulnerability assessment program there is no amount of studying or training that can take the place of real world experience. Because of the complex nature of this type of activity companies traditionally fail in implementing a vulnerability assessment program. For this reason many companies either do nothing because they do not have the expertise or they implement the program poorly.

| Year | 2000 | 2001 | 2002 | 2003 | 1Q-3Q 2004 |
|---|---|---|---|---|---|
| Vulnerabilities | 1,090 | 2,437 | 4,129 | 3,784 | 2,683 |

The first scenario is akin to claiming that because there are no mechanics in a car the flat tire can never be fixed. What is missing is a good plan to guide adept individuals in the implementation of a successful Assessment Program. Many companies actually have Security professionals but find that their security

professionals do not have the real world experience to define and implement a vulnerability assessment process. Faced with the realization that some type of vulnerability assessment must take place and lacking the expertise companies can become frustrated. Not having qualified professionals to assist in reducing risk will not stop an attacker from exercising a vulnerability and will not be a sufficient reason in a court of law or to shareholders should that malicious activity be successful. Honore de Balzac once stated, "It is easy to sit up and take notice, what is difficult is getting up and taking action".

The second scenario is indicative of poor planning or the lack of an organized and defined approach. The solution to this scenario is the same as the first; a good plan for implementation will ensure the successful execution of a vulnerability assessment program. An organization that has taken the initiative of implementing a vulnerability assessment program typically finds that either too much or too inaccurate data has been generated. Many organizations use their vulnerability processes to chase vulnerabilities within an organization without having defined baselines or without a good effective plan for execution. The following citation indicates the problem with using a vulnerability assessment process to chase vulnerabilities within an environment and to illustrate that preventative measures are more practical in vulnerability reduction; proving the adage that an ounce of prevention is worth a pound of cure. The total number of vulnerabilities report by CERT since 1995 is **15,629** (CERT Vulnerability Statistics). This being said, in order to assess 150 devices within an organization for all known vulnerabilities it would require approximately 2.35 million events. It can be interpolated that huge economies of scale can be made by reducing the number prior to those devices being implemented into production. This allows us to deduce that the reason to assess a device is not to identify its vulnerabilities but to compare it against our expectations. The following is the description of the rationale for conducting a vulnerability assessment by the NOAA Costal Center (specifically regarding disasters):

> "All communities are vulnerable to hazards. Your goal is to establish a starting point on the way toward reducing your vulnerability. A vulnerability assessment can be your guide for developing mitigation strategies and prioritizing mitigation projects. You should also plan to repeat the assessment to measure the effectiveness of your mitigation activities at appropriate time intervals for your community." (NOAA Costal Center)

This dictates that the primary and secondary purposes of a vulnerability assessment are not to identify all issues within an environment but to provide us a comparison against our baseline in order to certify implementations and to identify the presence of newly identified vulnerabilities. In order to accomplish these two goals we must identify the most effective method or steps for implementing this process of comparison and identification.

## *Vulnerability Assessment Steps*

The steps of the vulnerability assessment process are kept at a minimum to reduce confusion and to be effective. We must first identify the prerequisites to commencing a vulnerability assessment, secondly the steps or order of the scanning process, and finally the expected outcomes or actions. The expected outcomes will be addressed as each step of the scanning process is defined.

Prior to completing a scan certain potential issues must be taken into account. Primarily these issues revolve around how to identify what prerequisite measures should be taken to reduce potential undesirable effects this process may have on the production environment. As with any new process or device it must first be tested and validated for functionality. The testing for a product to conduct a vulnerability assessment goes much beyond standard User Acceptance Testing. A vulnerability assessment tool must be constantly evaluated because of the consistent updates to its vulnerability testing list (plugins) and the selective nature of the configurations or profiles. Before utilization of a vulnerability assessment tool for the first time and after every update or change to the tool a test of functionality must be conducted. These evaluations should run against a test environment defined for the testing of other production implementations. This environment should be representative of the existing production environment. It is preferred that the test environment be as closely related to the production environment as possible in order to provide insight to a devices operative response to a vulnerability assessment. This is extremely important because of the potential for service interruptions and other unexpected events even if all known precautions have been taken. For example, Michael Rowton, the author of a definitive tutorial on Nessus states:

> "As mentioned previously, you should always test new scanning preferences on non-production devices. The author of this tutorial has crashed several production servers by not following this advice (even with safe checks enabled, and no dangerous plugins enabled)." (Rowton)

By testing the vulnerability assessment tool in a test environment it allows multiple things to be accomplished. It not only allows the obvious, being the prevention of unexpected behavior, but also allows the professional to gain experience with the tool in a controlled environment. This is very important in learning to identify what False Positives and False Negatives to expect within the environment. The knowledge, recognition, and elimination of these are very important when defining an action plan and are best identified within the controlled environment of a lab/test environment. Upon validating expectations and usage of the tool within this controlled environment it is important to know exactly what procedural steps to take when running a vulnerability assessment tool within a production environment.

Production environments are made up of multiple different groups or classes of

systems but typically these can be grouped into one of two major groups and both must be addressed when identifying the steps or order of the scanning process itself. These two groups are critical infrastructure devices (servers, networking equipment, firewalls, etc..) and non-critical infrastructure (desktops, laptops, etc…). This was referenced above in the discussion of baselines in an earlier section of this paper. It is important to understand these two classes of devices because they will dictate the type or order of scans to ensure efficiency. This efficiency is based on the expected outcome and typical resultant actions to be taken upon those devices evaluated. If it can be assumed that the following are the most logical methods by which vulnerabilities can be addressed and represent the most effective order: remove service, disable service, substitute service, filter service, patch/upgrade service, or accept risk; an order of efficiency can be defined. For example, it would be fruitless to patch a service that should be removed from a device, or to filter telnet to a device that should have it disabled or substituted with SSH.

Beginning with the critical devices it is important to realize that two outcomes are expected. The first is to ensure that these devices match existing baseline configurations and the second is to ensure that once these baselines are validated any existing vulnerabilities are identified and remediated. This is broken into these two sections based on the most common methods for addressing vulnerabilities as listed previously. This means that the first 3 methods mentioned effectively remove the risk entirely while the second set of 3 controls address compensating measures for risk reduction/acceptance while leaving the service in place. Typically the comparison against a baseline can be done with service scans and the vulnerability identification would be done utilizing a vulnerability assessment tool. While both of these should be conducted they should be done in this order as previously mentioned. For example, it would be frivolous to run a vulnerability assessment to identify where the latest Sendmail patch is needed if there are only 3 devices within the entire infrastructure that should house this service. It would be more efficient to scan the entire infrastructure to identify that no new devices have had Sendmail implemented on them and address those that have by removing the service, disabling the service, or adding the device to the list of authorized devices (baseline inventory). This is easily accomplished utilizing service scanning tools that have the ability to either capture banners or identify services by finger prints, such as nmap. By only executing a service scan rather than a vulnerability scan at this point we reduce the amount of network traffic generated and the potential that a poorly written vulnerability assessment plugin will cause unexpected actions within the infrastructure. We then are able to address those devices that were known to be running Sendmail and those that were previously unknown. Dan Barker, a Security Systems Engineer with Inacom Information Systems points out that, “The awareness of what devices are currently on your network is considered the first tenet of patch management best practices. One must know what exists on their network before assuming the success of a patched

environment." (Barker) One would expect that since we have now identified the devices for which the Sendmail patch is destined the next step is to conduct a vulnerability assessment within the organization. This would not be effective at this point. Why scan an environment to identify where a patch is needed if, compared against our baselines, we have already determined the patch is required for Sendmail servers? This being said, the next step would be to apply the patch to the Sendmail devices within our environment and then follow up the patching with the vulnerability assessment to ensure that the patches have been applied correctly. Based on the scenario that we have just covered a couple of logical conclusions can be made.

- It is important to know what devices are critical within the infrastructure and what services are running on these devices.
- Service scans are more efficient in the identification, tracking, and resolution of potential security issues than an actual vulnerability scan.
- Vulnerability assessments are a tool to be used to validate where security measures have been implemented within an organization.

For this reason, broad based service scans, meaning a large range of services, should be executed on a fairly regular basis within a production environment in order to identify configuration or service changes on critical production devices. This is repetitive process unlike the ad-hoc service scan conducted in the Sendmail example above. Depending on the prevalence of change within an environment this may reflect a need for a weekly, bi-weekly, or monthly scan. The more change that is experienced within an environment the more need there will be for regular service scans. None of these deductions should imply that a full vulnerability assessment should never occur against critical infrastructure, only that the need for a full vulnerability assessment is limited. Using an example identified earlier within this paper, if we were to scan a critical infrastructure of 150 devices for all potential vulnerabilities identified since 1995 we would be generating millions and millions of network connections to critical production devices. This being said, a full vulnerability assessment against critical infrastructure should be done infrequently (i.e. – quarterly to biannually) not to identify vulnerabilities within the environment but to validate security controls; as well as, to ensure that changes made to production devices, while validated in a test environment prior to implementation, have been doubly checked to validate they have not effected the security posture of a device. This full vulnerability assessment should be done only after baselines have been identified and devices have been hardened. Failing to accomplish this after the baselines have been defined and implemented within an organization could result in the reporting of potentially thousands of vulnerabilities. These thousands of vulnerabilities or action items would create a tremendous amount of confusion regarding direction and prioritization with those tasked with the remediation of findings. An example put forth by an article published by Network Partners Inc. gives a great explanation as to the importance of

baselines and vulnerability assessments:

> "If we take a recent study of a simple internet gateway comprising 17 systems, it showed that installing every update, upgrade, fix and service pack would require approximately 1,300 patches in total over a 12-month period. Installing five patches every working day on 17 servers requires almost complete dedication of resources. With 1,700 servers, the task becomes formidable." (Network Partners Inc.)

Why send personnel to patch and monitor 100 devices running finger when the organization could just disable finger on all devices as a result of defined baseline or hardening guidelines?

The second group of devices that we must address is the non-critical devices within the infrastructure. These are separated from the critical devices because there is one additional notation to be made when conducting an assessment of these devices. The first step here is to conduct a broad range service scan of the non-critical environment, again on a regular basis. This is, as with critical devices, to identify configuration or service changes but also to identify non-critical devices that may be misclassified. Ideally a critical device would begin its life-cycle as a critical device but in many organizations this is not always the case. Devices may start as a non-critical device and as these devices transition from non-critical to critical they need to be identified, inventoried, and baselined as the critical devices that they have now become. This will also alert assessors to users that have set up services that should never be found on non-critical end user devices, such as: web services, mail services, ftp services, etc… Users should always be required to utilize approved and hardened servers that are maintained by administrative personnel. Users that configure their desktop devices with server class services provide another vector for attack for malicious users and code. For example, the Welchia worm took advantage of both a client RPC DCOM vulnerability associated with Windows XP but also a WebDav vulnerability associated with the IIS Web server. (Symantec Security Response Alert) This is a good example of how a desktop vulnerability while patched for the RPC vulnerability can provide a threat vector because there was no awareness that this desktop device houses a web server. The next step after addressing any issues identified from the service scan would be to conduct the

| Date | Action | Environment |
|---|---|---|
| January 15th | Service Scan | Critical |
| January 30th | Service Scan | Non-Critical |
| February 15th | Service Scan | Critical |
| February 30th | Service Scan | Non-Critical |
| March 15th | Service Scan | Critical |
| March 30th | *Vulnerability Scan* | Non-Critical |
| April 15th | Service Scan | Critical |
| April 30th | Service Scan | Non-Critical |
| May 15th | Service Scan | Critical |
| May 30th | Service Scan | Non-Critical |
| June 15th | *Vulnerability Scan* | Critical |
| June 30th | Service Scan | Non-Critical |

vulnerability assessment. This assessment, as was the critical vulnerability assessment, would be on an infrequent basis for the same reasons as the critical infrastructure.

It is important to define a formal schedule for conducting these assessments of the organization while still maintaining the ability to conduct ad-hoc scans upon the issuance of a critical alert. It is also important to conduct the recurring assessments of the critical and non-critical environments at different intervals to ensure that those individuals tasked with remediation are not overwhelmed with action items that require attention. For example, based on the time lines defined above a recurring schedule for the first 6 months of a year may look like Table 4.

***Table 4***

Creating a formal and repetitive process out of the Assessment Process is as important to instituting a successful and efficient vulnerability assessment process as are baseline configurations. Having defined both the order of a vulnerability assessment process and their application it is important to understand what ongoing maintenance is required to ensure that this process continues to function and is successful as an ongoing program.

## Ongoing Maintenance

| Date | Action | Environment |
|---|---|---|
| January 15th | Service Scan | Critical |
| January 30th | Service Scan | Non-Critical |
| February 15th | Service Scan | Critical |
| February 30th | Service Scan | Non-Critical |
| March 15th | Service Scan | Critical |
| March 30th | *Vulnerability Scan* | Non-Critical |
| April 15th | Service Scan | Critical |
| April 30th | Service Scan | Non-Critical |
| May 15th | Service Scan | Critical |
| May 30th | Service Scan | Non-Critical |
| June 15th | *Vulnerability Scan* | Critical |
| June 30th | Service Scan | Non-Critical |

Having defined both the Baseline and the Assessment process there must be a discussion on how to maintain these processes to ensure that they remain effective. In order to maintain these processes there are direct actions required to ensure a continued functioning and reliable Vulnerability Reduction Program. Therefore, when defining an Ongoing Maintenance Process 4 things must be addressed: post vulnerability scan documentation, baseline configuration maintenance, updating vulnerability assessment tools, and change control.

### *Post Vulnerability Documentation*

After successful completion of the first full cycle of the vulnerability assessment process, it is important to generate documentation as part of an ongoing

maintenance program. This documentation would reflect, for future reference, those detailed notations made during and after the scan process was completed. There are many things that will need to be documented during an assessment, as well as, upon completion of the assessment. It is important that during an assessment certain notations are made regarding the type of scan, the scope of the scan (devices and service/vulnerability configuration/profile), and the length of time each scan required to complete. While these are all important for historic reference, the length of time required for the scan to complete is the most important for heuristic purposes. If scans begin taking increasingly longer or shorter amounts of times this needs to be accredited to some specific action, such as the addition/removal of devices, changes to the assessment profile that may need revision, misconfigured network infrastructure, etc… Any issue that alters the behavior of a scan from what is normally experienced should be investigated to ensure that continued accurate data is collected and nothing is missed in the Assessment Process. Many things must be documented after a scan has completed. The post scan documentation should annotate what actions are expected to be taken by those personnel tasked with remediation, what changes to the infrastructure have been noted by personnel conducting the assessment, and any False Positives/Negatives that were noticed. The first two notations provide an ad-hoc tracking mechanism and the last notation assists with accuracy. Because some action must be taken on each item identified during a scan that does not match baseline configuration it is very important that these actions are tracked so that nothing is overlooked. An identified issue should never be left as unaddressed even if the action is to accept the risk and modify the baseline configuration with an exception. False Positives and False Negatives may indicate a need to update the vulnerability or service scan profiles to ensure that the data collected is accurate. It is imperative that only those actions that must take place and the supporting data be distributed to those personnel responsible for remediation to ensure clarity of direction and to maintain the Assessment Process's credibility. Any action for which an exception has been made and any False Positive that has been identified should not be distributed where possible as an action item. Many times this requires the data to be manipulated prior to distribution. Sometimes when addressing the action items provided formal changes are approved that may modify the standard build or hardening guidelines defined for the organization. This is an important distinction from an exception but both require a modification to existing documentation.

### *Baseline Configuration Maintenance*

Baseline configurations must be maintained as one of the most critical pieces of the Vulnerability Reduction Program. Any time a baseline configuration is modified an evaluation must take place to ensure that this modification does not put the entire organization at risk. This modification could be the addition of a control previously missing or the modification of existing controls based on other risk management criterion. This modification can or would apply to all devices

for which this baseline is applied. As stated, this is much different than a single exception to a baseline. Often these exceptions are made on a device by device or a service by service basis rather than against an entire device class (e.g. – all web servers, Sun servers, etc…). Any exception would indicate that a risk has been identified but for business reasons it can not be addressed at this moment in time and that there are no compensating controls efficient to address this risk. This is a decision that typically would be made at a managerial or executive level. It is a poor practice for system/security administrative personnel to begin accepting risks for an organization. Once this risk acceptance is documented it should be filed and dated and, if possible, an expiration date should be placed on this exception. Modifications or additions to a baseline may come from new projects or initiatives within the organization but many come as a result of updates to new vulnerabilities identified from the Assessment Process itself.

### *Vulnerability Tool Updates*

Updating the vulnerability process may be reflected in one of 3 ways: updating the scanning (service/vulnerability) profiles, updating the vulnerability listings (plugins), or creating assessment exclusions. Updating the scanning profiles is the easiest of the 3 and typically is made as a result of infrastructure changes to the organization or in the defining of what services/vulnerabilities will be evaluated during the scanning process. This is typically done to ensure that the process encompasses all areas of the organization. This should be documented as part of the standard documentation process but may not require a formal test of the new profiles, especially if they are modified as a result of infrastructure changes.

The second update does require testing but because it only affects the vulnerability scans and not the service scans it is more manageable. Service scans typically only have profile changes and those are rare, outside of infrastructure changes. The vulnerability scan updates do require testing because they modify what network traffic is sent to production devices. As previously mentioned all changes or updates to vulnerability assessment tools must be tested within an environment prior to updating the scanning profiles to include these vulnerabilities. It is also imperative that a review of the updated plugins be conducted prior to evaluation in the test environment. This is important to ensure that the new plugins are understood and that any potential False Positives or False Negatives are identified and noted.

The third is something that should be well documented and very rarely used. This is the alteration of the scanning profile to exclude specific devices that have ongoing difficulty with the scanning process. Many times this is seen as an exclusion of devices that experience service interruptions during the scanning process, such as: Mainframes with older IP stacks that fail during a service scan, tftp services that fail during a udp scan, etc. These should only be excluded for a short time period to allow for investigation and resolution. The

exclusions are made so as not to delay the entire Assessment Process but to ensure that business processes proceed during these assessments. Each of these notations and modifications was made at the administrator level but this does not indicate that some form of formal change control should not be utilized when evaluating and updating these processes.

### *Change Control*

Change control is one of the most important formal procedures that exist within an organization that hopes to maintain a stable environment over an extended period of time. While change management may be viewed by some as a hurdle to productivity, well implemented change control will prevent frivolous change within an organization and ensure that all changes are communicated amongst all system stakeholders. In the Eight Rules of Security as defined by SilverStr, change management is defined as a principle tenant of security:

> "When you make a new change you expose your business to new risk. Any time a change is to occur you must consider all possible security implications. You MUST have a clear and concise change management process that you adhere to. To remain secure you must be aware of changes going on within your environment, and what impact those changes have on you." (SilverStr)

Change management is used to submit for review and obtain approval for changes to be introduced into an environment. This is a primary outlet for the tracking of changes and the generation of historical reports to identify where changes have altered a device from its baseline configuration. Any change made to a device that is not documented within the change control process should be viewed by executive staff as a failing in this principle tenant of security. Any significant change to devices should be evaluated for security controls and the need for an ad-hoc vulnerability assessment (service scan and vulnerability assessment) to ensure all required security implementations, required by the change, have been implemented. Change management is also the ideal mechanism by which to track changes or actions identified as a result of the Assessment Process. This mechanism allows scheduling, notations, and hurdles to be identified in a formal manner and raised to the level of executive management if there are significant issues with addressing identified risks. According to Fred Nickols at Distance Consulting a good change management program will provide an "organized process for getting from one [state] to the other" and that successful organizations identify at an early stage how change management structures will appear and their implementation. (Nickols) Mr. Nickols also points out that organizations typically survive those individuals that establish them and that change management is one of the only mechanisms to ensure that some continuity is maintained during these transitions within the organization itself. (Nickols) For this reason change management must be implemented in a manner that will allow survivability of the processes and

procedures beyond those that implement them. Meaning that if a new group of individuals are tasked with either the Assessment Process or the remediation it should be easy to identify all of those state changes that have occurred and all of those that are currently being evaluated and executed. Mr. Nickols states that a good change management program should be much like the formation of the U.S. Navy, "It was designed by geniuses to be run by idiots" (Nickols). While a statement such as this typically should not find its way into a formal research project, it is indicative of the need for a well planned but simply executed change management program. Administrators should not find the change management process so arduous as to be a hindrance to ensuring that the process is followed. Defining a change management process is beyond the scope of this document but a great deal of information can be found on the Internet at locations such as http://www.change-managment.org. Having discussed how to begin, execute, and maintain a Vulnerability Reduction Program a Vigilance Process is an easily employed yet important byproduct of the implementation of these vulnerability reduction processes.

## Vigilance Process

As ongoing support for the Vulnerability Reduction Program it is important that an organization define what methods it will used to track and generate alerts based on emerging threats and newly identified vulnerabilities. At a high level, defining a vigilance practice must take into account what mechanisms will be used to identify emerging threats and newly identified vulnerabilities, how to evaluate them, and what actions to take against these threats.

### *Identification*

Identification of emerging threats and vulnerabilities must be a well measured process and inclusive of all areas within the organization. These processes will reflect each class of device for which a baseline is defined; keeping in mind that a representative baseline should exist for every functional device present in an organization's infrastructure. For example, in an organization that has HP-UX/Apache, Win2k/Apache, WinXP Desktops, and Win2k/IIS6.0 mechanisms must be in place to identify new issues with each OS and application. This mechanism may be email alerts from the vendor (Microsoft, HP, etc..), emails from a public alerting service (CERT, Securiteam.com, etc..) or a paid alerting service. Many organizations shore this alerting process with Intrusion Detection Systems for real time attack threat alerts. What makes these processes efficient is the baseline and service scan process. Using an earlier example, if an alert is issued for finger and, based on our baselines and our service scans, we can determine that finger is not represented within our environment this alert can be deferred regardless of the urgency annotated by the alerting mechanism.

The IDS alerting is also made efficient because the Alerting would be made specific to those services and operating systems as defined by the baselines and the service scans. If a finger attack was to originate externally against our environment it would be logged within the IDS but would not generate an action alert that requires a response. Based on those services and operating systems present within the environment a defined "watch" list can be created so that only those alerts that apply will be addressed. It is important to evaluate the alerts that do match the defined watch list within a timely manner.

### *Evaluation*

Evaluation of these alerts should also be done in a formal and defined manner that is not restricted to only one small group of security individuals. The evaluation should be done by a group of individuals that have the ability to take into account both the business impact and the security impact of instituting the actions as warranted by the alert. In many organizations this group of individuals would be the same as the group responsible for the CERT (Computer Emergency Response/Readiness Team) or CSIRT (Computer Security Incident Response Team) process. Based on information within the alert, the business impact, and existing compensating controls a schedule to address the alert should be defined and tracked via appropriate change control mechanisms.

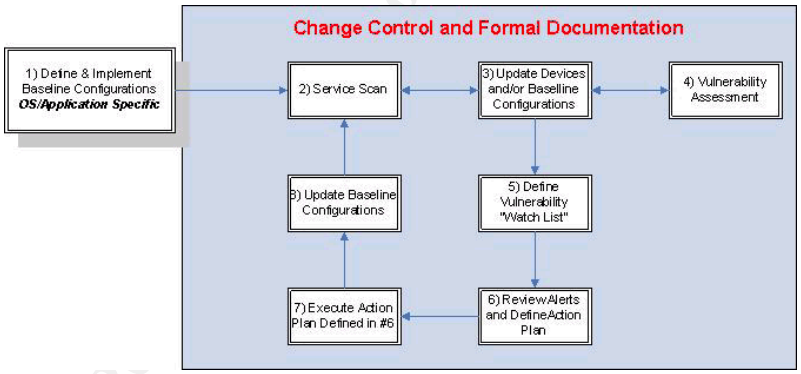
### *Actions*

The actions taken to address these alerts are varied but typically these actions will be the same as those identified with the "Vulnerability Assessment Steps" of this paper. These actions will typically be to remove/disable a service or function, implement compensating controls (filter/substitute), apply a patch, or accept the risk. This section is not meant to define all of the methods available for addressing a risk but to annotate that each area will require some formal method as listed above in this paper. If a service is removed, disabled, substituted, or patched a formal change process must be executed to document the change and the Baseline Configurations must reflect these actions. Any risk acceptance must be well documented and tracked as previously discussed. Any action taken must be well documented and applied to processes previously discussed in this paper, which in turn cyclically modifies what the vigilance posture looks like.

From this point it is simple to associate a vigilance practice with a Vulnerability Reduction Program and is a natural progression in effectively identifying, alerting, and addressing emerging risks and vulnerabilities to an organization. This represents the final stage in what ultimately is a self-feeding Program.

## Conclusion

This paper has addressed how to effectively implement a Vulnerability Reduction program in an efficient manner. It outlined the importance of Baseline Configurations as a critical prerequisite prior to ever executing a vulnerability assessment of the environment, the most effective steps by which to conduct the Assessment Process, the Maintenance Processes needed to ensure this Program remains efficient, and the Vigilance Process as a beneficial byproduct of this Program. On a consistent basis it was noted that many of these processes feed other processes within this Program. If there is a failing any of these processes it will affect the ability of each of the other processes to function adequately. As stated, this Program is a very cyclical one and each phase is dependant upon accurate documentation and maintaining up to date processes. Because a major security axiom is to "Know Thyself" it is crucial that these processes be enacted repetitively to ensure that a constant awareness of the infrastructure is maintained. In the end all of the steps in this Program fall back to the principle of awareness. It is impossible to adequately address a risk that is unknown. The plan that is outlined within the paper was meant to address the high level issues and requirements in implementing a Vulnerability Reduction Program and any detailed implementation controls fall outside of the scope of this paper. There are many resources available on the Internet to provide guidance on how to specifically handle each of these steps but detailed guidance will be dependant on the toolkit chosen by the organization to implement these processes. This plan is not only to offer guidance to those organizations that have dedicated security personnel in the hopes of implementing a Vulnerability Reduction Program, but is also meant to enable those organizations that currently do nothing with regards to Vulnerability Reduction. The most critical step of this entire Program is the recognition that one is needed and taking initiative to begin implementing it. In the words of George S. Patton, "A plan violently executed now is better than a perfect plan executed next week".



# References

Tippett, Peter Dr.. "Keep it Simple." TruSecure Corporation. 25 November 2002. 19 December 2004.
<http://www.trusecure.com/cgi-bin/download.cgi?ESCD=W0073&file=doc611.pdf>

Miuccio, Bert. "Research Report Summary." The Center for Internet Security. 15 November 2004.
<http://www.cisecurity.org/Documents/research%20-%20case%20studies.ppt>

"CERT/CC Statistics 1988-2004". CERT Vulnerability Statistics. 19 October 2004. 15 December 2004. <http://www.cert.org/stats/>

"Security Knowledge in Practice". CERT Security Improvement Modules. 30 May 2001. 30 November 2004. <http://www.cert.org/security-improvement/practices/p068.html>

"CVE Candidate Vulnerabilities". NIST ICAT Vulnerability Statistics. 12 September 2004. 17 December 2004. <http://icat.nist.gov/icat.cfm?function=statistics>

Feldman, Jonathan. "No Default Deny? Disaster!." Network Magazine. 15 September 2004. 26 November 2004. <http://www.networkmagazine.com/showArticle.jhtml?articleID=47208578>

Rowton, Michael. "Introduction to Nessus Tutorial." SecurityDocs. 22 November 2004. 12 December 2004. <http://www.securitydocs.com/library/2730>

Vulnerability Assessment Tutorial. 2003. NOAA Costal Services Center. 25 November 2004.
<http://www.csc.noaa.gov/products/nchaz/htm/tut.htm>

"Why do a Network Vulnerability." Network Partners, Inc.. 30 November 2004.
<http://www.routers.com/WhyVA.html>

Perriot, Frederic and Knowles, Douglas. "W32.Welchia.Worm." Symantec Security Response Alert. 19 August 2004. 15 December 2004.
<http://securityresponse.symantec.com/avcenter/venc/data/w32.welchia.worm.html>

Barker, Dan. "A Good Patch Management Strategy." Inacom Information Systems. 2004. 5 December 2004. <http://www.inacom.com/display.aspx?page=/newsletter/patch.aspx>

SilvrStr. "The Eight Rules of Security." SilverStr's Blog. 29 December 2003. 12 December 2004.
<http://silverstr.ufies.org/blog/archives/000468.html>

Nickols, Fred. "Change Management 101: A Primer." Distance Consulting. 2004. 13 December 2004. <http://home.att.net/~nickols/change.htm>