Implementing Vulnerability Assessment with eEye's EVA Suite – Case Study

By Kevin Austin
January 12, 2004
GIAC Security Essentials Certification
Version 1.4b, Option 2

Abstract

Vulnerability assessment is an important part of any Defense in Depth implementation. I discovered that in my company vulnerability assessment was not being used to its full advantage inside the perimeter. My team was continually fighting the same battles against unpatched and vulnerable systems as they would acquire various viruses from the network. The product that I was using to evaluate our environment did not allow us to schedule scans, and I could not afford a license that would allow us to scan our entire IP range. It was decided that we needed a true enterprise solution that would allow me to evaluate our entire environment on a regularly scheduled basis.

After looking at several vulnerability assessment products I finally found one product that met all of our needs, and was affordable enough to fit into our budget. Once I tested and rolled the product out to production I was able to schedule scans of our many different environments, and gather much needed information about the computers in those environments. This allowed me to have a better picture of how vulnerable we were as a company to viruses and other malicious activity. Once I was aware of the issues I was better able to address them quickly and efficiently. A process was put in place to schedule scans with audit policies that I had tailored to our environment. Anyone who has looked at an initial report from a vulnerability scanner will tell you that only half of what is discovered will be important to any one environment. Therefore, it was important to configure the scans and reports so that they would only reflect information that was relevant to my company's environment. The time spent deciding on an appropriate vulnerability assessment product, and then using that product with custom tailored scans has helped to improve the overall security of the company. I am now able to quickly get an overall picture of the outstanding vulnerabilities, and track the patching process of those vulnerable computers in a timely manner.

Before

My company's business is retail sales, and a large percentage of those sales are done over the web. Therefore, it is very important that our servers not be susceptible to the multitude of viruses and operating system vulnerabilities that are being exploited on the internet. The problem was that I really had no reliable way to verify that all of our servers had been patched and hardened against all of these vulnerabilities. I did have a license for a vulnerability scanner, but it was not a true enterprise solution. The scanner that I had been using did not give me the ability to scan our entire environment due to the limited license. Another issue was that we have segregated several of our environments with firewalls, and I had seen issues scanning across the firewalls previously due to intrusion prevention software built into the firewalls. The intrusion prevention piece of the firewall would deem a majority of the traffic from the vulnerability scanner as malicious activity, and would then drop the offensive traffic. Another issue I found when scanning across firewalls was that if the scans were running too quickly the traffic could sometimes be overwhelming, and cause a network bottleneck. This being the case, I could not get reliable information when scanning across our internal firewalls. Therefore, I wanted a solution that would allow me to manage distributed scanners in all of our environments, including DMZ's and staging environments. All final decisions were made by the Information Security Team as to what products to test, and who would finally get our budget dollars. I however, was tasked with installing and testing all evaluation products and then putting together a list of pros and cons for the decision making process. Once a vulnerability assessment product had been decided on I was the primary technical person for all rollout duties and issues.

My company has a fairly large and complicated server environment, and the fact that I was unable to fully audit this environment quickly became an issue due to several Microsoft vulnerabilities being discovered every month. The fact that I had no reliable way to monitor and audit the vulnerabilities on our servers put us at high risk for virus infection and other malicious activity. It was also troubling to me that some analysts like Gregg Keizer[1] are of the opinion that 2004 will be even worse than 2003 as far as computer security issues are concerned. As if 2003 wasn't bad enough, now I had to think ahead to 2004 and all the potential issues that would continue to be found in Microsoft operating systems. Although there are firewalls in place to protect the servers from both internal and external sources, most recent worm type viruses have used well-known ports and create traffic that can be passed as "normal" by firewalls. In the article "A Comparison Study of Three Worm Families and Their Propagation in a Network," Daniel Hanson[2] analyzes worm propagation strategies, and the success of those strategies. In the article Hanson points out that "Organizations must begin to address the weaknesses that are inherent in the topology that has been commonly deployed in the past. The notion that there is a safe local network and a hostile external network is a misnomer." This issue became very apparent to us when the Welchia virus was released. Although 95% of all machines in the company had been patched, the 5% that had not been patched became an issue very quickly due to the way the virus propagated itself. Most likely the virus got in the building from a laptop which had acquired the virus on a home network, and was then brought into the building.

I would continue to run into these issues as long as I was not running scheduled scans of the environment on a regular basis. Without the ability to schedule scans and run delta reports on the findings of those scans I was constantly hunting down unpatched and sometimes virus infected computers that in some instances had only been on the network a number of days. With new machines coming online, and old machines being rebuilt, I needed a way to consistently audit all machines on the network and keep track of their patch status.

It was decided that we needed a vulnerability scanner that was more robust and that would be able to scale across our ever growing environment. At this point the issue became choosing a product that would balance both cost and functionality, while fulfilling all of our requirements. I also took into consideration the four points listed in the chapter on vulnerability scanning in the SANS GSEC course, which consisted of; flexible licensing, CVE standard support, delta reporting, and pretty executive reports[3]. Based on this our final requirements were:

1. The vulnerability scanner needed to have the ability to manage many scanners from a central location.
2. Information obtained from the scanner must be accurate and reliable.
3. I must be able to afford a license that would allow me to scan our entire environment, including international subsidiaries, and that license should be easily upgradeable.
4. The scanner must be able to run on the hardware that I already have in the budget.
5. All information from scans should be stored in a SQL database backend, and reports should be customizable to allow for trend analysis.
6. All vulnerabilities must be searchable and cataloged by type.
7. A remediation solution included in the product would be a plus.

Using these requirements I began working with several vendors on getting demo scanners so that I could evaluate their products in our environment. I worked with vendors like Foundstone, ISS, Eeye, and Saint. Testing many scanners also allowed me to test the consistency of scan results on a single subnet against one another. This would give me an idea of the accuracy of the scanners from each vendor.

The first product tested was from Foundstone. The Foundstone product had a web based management system that allowed many administrators to access the system from a web browser. The Foundstone product came pre-installed on two 1U servers. One server was used as a database backend and the other was the scanning and management engine. The one thing that made this scanner stand out from the rest was its remediation system. Once a vulnerability was found it could be assigned to an administrator for remediation. After the administrator had patched the server or otherwise fixed the vulnerability, he would log into the Foundstone interface and close the ticket. Once the ticket was closed the Foundstone scanner would automatically run a scan against the relevant machine and verify whether or not the vulnerability had in fact been patched. If the machine was still vulnerable the ticket would remain open until the issue was truly resolved. This feature would allow us to track the level of all vulnerabilities in the environment and their remediation process.

The next product I tested was the Saint Vulnerability scanner. The Saint product ran on Linux, and was a stand alone scanner with no centralized management of multiple scanners. Although this was a solid scanner which scanned quickly and had reliable results, it did not meet the needs of our enterprise. With a more intuitive interface, and distributed management Saint would be a solid enterprise scanner. The low price of the Saint scanner did keep it in the running as a secondary scanner for audit verification though. In our testing of multiple scanners I found that there is no vulnerability scanner on the market that can profess 100% accuracy, so I found it useful to sometimes scan with two products and compare the results. So, although I knew that Saint would not be the answer for all of our enterprise scanning, I did feel that it could still prove useful in our environment.

eEye Retina was another vulnerability scanner that I tested. The eEye EVA (Enterprise Vulnerability Assessment) suite of products used a web-based central management console to manage distributed scanners. This would allow us to scan without putting stress on our internal firewalls. All information retrieved from scans was stored in a SQL database. All production SQL instances in our environment are clustered instances, and it was important that if the product I chose used a SQL backend that it be compatible with a clustered instance. Retina had no issues during our test with the staging SQL cluster that was already deployed. Retina also had a remediation solution that allowed tasks to be assigned based on issues found during scans. I also found that in many cases eEye was the first company to put out a vulnerability check when a major Microsoft vulnerability was discovered.

The ISS Internet Scanner was the product that I had already been using in our environment. Internet Scanner had the ability to run in a distributed architecture as well, but addition licenses were required to use this functionality. Internet Scanner had been a reliable stand-alone product, but when I installed the demo distributed model I found it to be somewhat cumbersome and hard to manage. Another sticking point with the ISS product was that it was not cluster aware. This meant that on top of needing to purchase additional licenses I would need to buy another server for the SQL backend. Besides these reasons, the number one issue with ISS was the high price of licensing for our environment. I really wanted to be able to afford a robust enough solution that would meet all of our needs. Internet Scanner was not the product for us.

I tested products from these vendors for months and found that many of the products met most of our needs; however in the end the eEye Retina Enterprise Vulnerability Assessment (EVA) product was at the top of our list. I chose the Retina vulnerability scanner based on the fact that it met all of our criteria including a remediation solution that would allow me to automatically assign tickets based on issues found on scanned computers. I also liked the ease of installation and management of the Retina product.

During

Once I had decided on the eEye product I needed to decide how the scanners would be deployed in our environment. The suite of products that make up the eEye EVA can be architected a number of ways. The products that make up the suite are Retina, REM Event Server, REM Event Manager, REM Event Client, and Retina Remote Manager. Basically how these products work together is like this: Retina scans are kicked off remotely through the web based Retina Remote Manager. Events, which consist of all information gathered by the scans, are passed securely to the REM Event Server by the REM Event Client. There can be any number of REM Event Clients reporting events to a single REM Event Server. The REM Event Server then puts all gathered information into the database. REM Event Manager allows users to administer all of the REM events from a centralized web interface. Within this interface reports can be created as well as all other administration of the product. By creating separate programs for each function of their EVA, eEye is able to scale across very large environments and still keep a central repository of events. This image from the EVA Installation Guide[4] should simplify what eEye calls the "Advanced EVA Network."

I used this network architecture in our environment, except I put both the REM Event Server and the REM Event Manager on the same server. I used the database cluster that was already built in the production environment. Knowing that I did not want to scan across any of the internal firewalls I decided that a Retina scanner should be deployed in each of the relevant environments. This included the development, staging, production DMZ, corporate (including International subsidiaries, desktops and corporate file and print servers), and a scanner in the lab environment for testing. I wanted a physical server for the REM Event Server\Manager, the corporate scanner, and the production DMZ scanner. In the spirit of cost savings I decided that for the lab, development and staging environments virtual machines (VMs) would be sufficient to handle the scanning load due to the smaller sizes of these environments. With a VM server already deployed in these environments it would be easy to get these scanners up and running quickly and they would perform basically the same as the physical servers. Figure 1 (All Figures are at the end of the document) shows what our production architecture looked like once the rollout was complete.

Since I had already tested the product in a lab environment, I was ready to move forward with the production rollout. As soon as I received the licenses, and all hardware was prepared with the operating systems hardened to corporate specifications, I was ready to begin the production rollout. All IIS installations were locked down with Microsoft's IISLockdown tool, in addition to all relevant patches being installed on the servers. Although this paper is not meant to be an installation manual for the eEye EVA suite, I will briefly highlight the installation process to better demonstrate the relationship of the different products to one another.

The first thing to do was to create the database in the production environment. Included with the product is a SQL script that creates all database tables for the REM Event Server\Manager. Once this had been done and the SQL user had been created and given the proper rights to the database it was time to install the REM Event Server.

The REM Event Server uses a System DSN for database connectivity, which needed to be created on the server. Once this is done and connectivity is established with the database the installation continues. The installation then comes to a certificate creation portion of the install. This is where the Event Server creates an SSL certificate for the secure communication between the Event Clients and the Event Server. A client certificate is then exported. This certificate is needed on all Event Clients, and the Event Client install will prompt for the location of this key and the password to that key. As stated in the EVA Installation Guide, it is very important to store this certificate in a secure location, because an attacker could use this key to intercept REM security events.

Next I installed the REM Event Manager since it would be running on the same server as the Event Server. Since the Event Manager is a web-based management system, the installation prompts the user for website configuration, database location, and administrator password. Once this is completed and the license is installed the interface is ready for testing. Figure 2 shows the login and the resulting interface once the installation was successfully completed.

Now that the REM Event Server and the REM Event Manager had been installed and connected to the database it was time to setup the Retina Vulnerability Scanners with Retina, REM Event Client and Retina Remote Manager. All three of these products, Retina, REM Event Client and Retina Remote Manager need to be installed on all servers which will act as scanners. For my environment that was five servers. Fortunately the installation of these products is very straight forward and I didn't run into any issues. First I installed Retina on the servers. This installation is point and click, and there is no configuration to be done by the user other than license key installation.

The next product installed on the scanning servers was the REM Event Client. The installation is simple enough, but there is some configuration to be done once the installation is complete. The first thing that is asked in configuring the Event Client is the IP of the REM Event Server, and the port used to connect to the server. The port number can be selected during the Event Server installation, or left as default. Next you are prompted for the client certificate which was created during the installation of the Event Server. After browsing to the appropriate directory for the client certificate, and entering the correct password you will see the dialog asking for REM Event Aware Products. Here you will check off Retina as a REM Event Aware product. This will tell the Event Client that all events coming from the Retina program on this server will be forwarded by the REM Event Client to the REM Event Server. Once connectivity has been tested and events are being passed to the Event Server the installation of the Event Client is complete.

The third and last program to be installed on the Retina Vulnerability scanners is the Retina Remote Manager. This allows the user to kick off Retina Vulnerability scans and manage those scans for that particular scanner through a web-based interface. Once again the installation is easy and straight forward enough, but there is still some configuration to be done before we are through. Just like in the Event Manager installation, the user is prompted for website configuration. The default website can be selected, or another site if there has been one created on the server for Retina Remote Manager. Once this is done the installation is complete.

The one undocumented issue that I did run into was NTFS permission problems when accessing the Retina Remote Manager website. After the initial install there is an access denied error when a user tries to access the Retina Remote Manager. I never found this documented in any of the eEye documentation, but all that needs to be done is to turn off anonymous access to the website and turn on integrated windows authentication. Since the only access granted to the website directories is to local administrators, then the only users that will be able to access the Retina Remote Manager Web interface will need to be administrators on the server. This is not an issue since the only people who will need to access the web interface, Information Security Team, and the Infrastructure Team, are already granted administrator rights on the servers. Although, further down the line it will be easy to get more granular about who has access to these web interfaces through NTFS and the user policies in the REM interface, I felt that this would be restrictive enough for the initial rollout.

After Retina, Retina Remote Manager, and REM Event Client have been installed on the servers, they need to be added to the REM Server interface. This is done under the Scanners portion of the REM Server interface. The only information needed to add the scanner to the interface is the server name, IP, and the path to the Retina Remote Manager website on the server. Once all scanners have been added to the interface it will look like Figure 3.

In Figure 3 you can see all five scanners have been added to the interface. This allows easy access to all five scanners by clicking on the Retina link next to the scanner name. Clicking on the Retina link will open up a new browser window which will allow you to manage the Remote Retina interface on the scanner you have chosen. This interface allows you to initiate scans, configure IP address ranges, schedule scans, and modify audits. Figure 4 is a screenshot of the Remote Retina interface.

Once all the scanners had been configured in the REM interface, I pulled all IP subnets that we were using in all of our environments from Active Directory. I then broke those subnets down as to which scanner I would be using to scan them with. From within the Remote Retina interface I used this IP subnet list to create address groups for each appropriate scanner. Some scanners, like the development and staging scanners, would only have one or two different address groups needed, but other scanners like the corporate scanner would need many more because it would be scanning the corporate office as well as all of the international subsidiaries.

After all scanners had been configured with the appropriate subnets to scan, and test scans had been run I needed to figure out what information I wanted to get back from my scans. From the test scans, which included all vulnerability checks, I could see that I needed to figure out which of these were of value to me. Of course, many of the vulnerabilities that were important to check for on the web servers might not be relevant for corporate desktops. I knew from previous test scans and the advice given in the SANS GSEC[3] course book that I needed to run heavy scans, but not so heavy that they would bring down production servers. I also knew that I might need to throttle down the scan process so that I would not cause network issues during scans.

I spent many days running test scans on all of the different environments using the default "complete scan" policy to discover which scan results were relevant and true issues for that environment. Initially there were many issues showing up on the scan reports that I did not deem important to report on. I wanted to focus primarily on the high risk vulnerabilities, and then move onto other issues as time warranted. By removing many of the informational audits along with low and medium vulnerability checks from the default policy, the scans were able to run quicker and the information gathered was easier to decipher. Since my company is almost purely a Microsoft shop, I focused the scans on all high risk vulnerabilities pertaining to Microsoft products. For all non-Microsoft computers I was able to discover and address them with the scanner's OS fingerprinting technology. I created a separate scan policy for all UNIX and other non-Microsoft computers so that I could report separately on those issues. I decided that since IIS is installed by default with Windows 2000, that there was a possibility that there were many IIS installations on computers in the desktop subnets. There are many web developers in our company, and almost all of those developers runs a server OS on their desktops for testing. This being the case, I felt it would not hurt to run the same scan policy against all desktops as I was running against the servers. Running the same scan configuration would allow me to discover all machines in the desktop subnets that had IIS installed and if those instances were vulnerable.

In instances where we were in the process of rolling out a critical patch for our Microsoft systems I would create a custom scan that would look only for that patch or set of patches. Scans with these policies would be run on more of an ad-hock basis, or on a daily basis once the patch deployment had begun. Retina allowed me to run delta reports on these scans so that I could see how the patch deployment was progressing. Of course these audits would also be added to the scan policies that had already been created. I also continued to run scans with the default, or complete scan policy on a weekly basis. From the REM Event Manager I could create reports that would only pull specific information I was looking for from a complete scan. For instance, this was important in case issues like the scanner not having the ability to access the registry of computers remotely. The Windows service that runs the Retina scanner runs under a Windows domain account. This account is in a group that should have local administrator rights on all machines in the company. Since a majority of the information gathered by Retina is pulled from the registry it was very valuable to be able to discover the machines that it could not access. Any machine that Retina could not access, and therefore not completely audit was a potential threat to our environment. With the ability to pull this information from the complete scans I was able to address the problem computers and remedy the issue.

As far as throttling the scan speed, I did not find any network issues while scanning any of the internal environments. Since I had placed a scanner in each of our internal environments I was not scanning across any firewalls. Previously the only time that I had seen network issues was when I had initiated scans across a firewall. International subsidiaries were another issue though. Since most of our international subsidiaries connect to our network through VPN they were coming across a firewall to get to our network. This firewall did not have the intrusion prevention piece previously discussed, so that was not an issue. There was however still a possibility that the scans would cause a network bottleneck if they overwhelmed the firewall. Since there would only be a few machines to audit in most of our international subs, they would have to be scanned across the firewall. I decided the easiest way to throttle the scans would be to limit the number of machines audited at once, and the speed which those audits were run. By default the number of machines audited at once in Retina is ten. To be sure there were no issues, and since there were only a small number of machines that would need to be scanned for any one subsidiary, I brought this number down to three. I also changed the speed from a setting of ten to five processes per module. This worked fine for scans that were run against the international subsidiaries, but the issue was that the same scanner being used for the subsidiaries was also used for corporate desktops and servers. The general settings which controlled the speed of scans were global for the scanner. That meant that I would either need to change the settings back for corporate scans, or create a separate scanner for the international subs. I did not want the corporate scans to run all day, and I also knew that changing the settings back to normal on the scanner would somehow fall through the cracks. So, I decided to create a separate Retina scanner for the international subs. Without the budget dollars for a separate piece of hardware I once again decided to use a virtual machine for this scanner.

Now that I had all audit policies created and tested in all the different environments I needed to complete the scheduling process. I decided that the scans using the custom audit and the full scan should be run only once a week each. The scans would be run on different days, so that I would be able to gather information on issues addressed throughout the week. Servers could be scanned at any time of the day or night, but I was aware that the scans had the potential of causing some issues with the servers[3]. In the article "We can't live in a risk-free world…" Andy Coote[5] analyzes the impact of information security on the business, and how the success or failure of those strategies can affect allocated budget dollars in the future. Since I obviously wanted to keep my budget dollars, and continue to enhance the security of my company in the future, I wanted to take into consideration the impact of my scans on the business when creating my scan schedule. Our heaviest web traffic is between the hours of 5:00 to 9:00 PM, and during busy periods traffic could be heavy until as late as midnight. Taking this into consideration I decided that the server scans should be run after midnight once a week. Since many of the desktop computers in my company are laptops that go home with their owners each night, I needed to run the desktop scans during the day to catch as many machines on the network as possible. I am still in the process of scheduling successful international scans, due to the many different time zones involved. Other than the international scan times, here is a matrix of the scanning schedule.

	S	M	T	W	T	F	S
Desktops		C 2:00PM				A 2:00PM	
Corporate		C 2:00AM			U 2:00AM	A 2:00AM	
DMZ		C 2:00AM				A 2:00AM	
Dev		C 2:00AM				A 2:00AM	
Staging		C 2:00AM				A 2:00AM	

C= Complete Scan A= Custom Audit U= UNIX Scans

This schedule would allow me to gather information twice a week, and formulate deltas on that information for reporting. As I said before, during critical patch implementation I would run scans more frequently.

Now that I had configured all the software, created the audits, and scheduled the scans the initial implementation was complete. The only thing left to do was to create pretty reports for the executives and informative reports for my team and the others working on resolving any issues found. With the ability to create custom reports easily in REM Events Manager, I was able to accomplish this in a short amount of time. Now I am able to quickly pull up reports and identify any particular vulnerability throughout the company with little effort. Although the remediation of any found vulnerabilities is out of the scope of this paper, I am currently working on automating task assignment based on events in REM Event Manager to individuals and teams for remediation of those issues. The eEye EVA suite will make this very easy, once I have established the lists of individuals and groups for assignment.

After

The decision to purchase the eEye EVA product suite was not an easy one to make with all the different choices in the vulnerability assessment market these days. The time that was spend looking at many different vulnerability scanners, and testing those scanners in our environment helped me to make a more educated decision. Figuring out exactly what I wanted from a vulnerability scanner, and then putting each product up against that list allowed me to easily weed out the products that would not work for our environment. The information gathered from the SANS GSEC class was also a great help in making a product decision, as well as during the implementation of the product that was chosen. Now that the eEye EVA suite is in place in my company, I am able to identify vulnerable computers usually before those vulnerabilities become and issue. When critical patches are released, I am able to track the progress of the rollout of those patches. With the REM Event Manager I am able to create reports that will give me a picture of how many machines are being patched on a scan by scan basis. The issue of chasing down vulnerable and infected machines has been simplified with the use of the EVA suite.

"Network-based vulnerability-assessment scanners play a critical role in the identification process by enabling their operators to spot security deficiencies before the bad guys do,"[6] and I feel that the implementation of this vulnerability scanner has enhanced the overall security of my company by lowering the number of unpatched computers that fall through the cracks. Finding the vulnerabilities is only one half of the equation though. The other part of the equation is the remediation of issues that are found. In the future I will be able to automate the assignment of remediation tasks based on Retina scans. Until then it is still a process based on reports sent out on a regular basis. Although we still have the risk of computers with vulnerabilities not being addressed immediately, I also have to take into account the business impact of installing patches on production servers. I think the fact that viruses continue to find ways to impact even the most secure environments shows that there is no way to be completely invulnerable to malicious activity. But, with the use of a vulnerability scanner on a regular basis, and then following up on found issues, I can sleep a little better knowing that I have done my due diligence.

Figures:

Figure 1.

Figure 2.


REM
REMOTE ENTERPRISE MANAGEMENT
Login
Manage Your REM Server
Managing Your REM Server
Use the tools and information found here to configure and manage REM.
Manage your account
Logout
Read about REM
Read about Event Administration
You can configure the following properties of your REM installation:
Quick Look
Events
Tasks
Reports
Scanners
This is the Scanners section of REM. Remote eEye engines can be managed and controlled centrally from here.
Manage scanners
Manage scanner categories
Read about REM Scanners
How to create an scanner machine
Rules
Users
Options
Local intranet

Figure 3.

Figure 4.

References:

[1] Keizer, Gregg. "Security Threats: Bad In 2003, Worse In 2004?" URL: http://www.securitypipeline.com/shared/article/showArticle.jhtml?articleId=17100252

[2] Hanson, Daniel. "A Comparison Study of Three Worm Families and Their Propagation in a Network." URL: http://www.securityfocus.com/infocus/1752

[3] Cole, Eric; Fossen, Jason; Northcutt, Stephen; Pomeranz, Hal. SANS Security Essentials with CISSP CBK. 689-750.

[4] eEye. EVA (Enterprise Vulnerability Assessment) Installation Guide. 3-4.

[5] Coote, Andy. "We Can't live in a risk-free world…" URL: http://www.scmagazine.com/scmagazine/2004_01/feature_2/index.html

[6] Novak, Kevin. "VA Scanners Pinpoint Your Weak Spots." URL: http://www.networkcomputing.com/1412/1412f2.html