An approach to the ultimate in-depth security event management framework

GCIH Gold Certification

Author: Nicolas Pachis, npachis@vt.edu

Adviser: Mrcorp@yahoo.com

Accepted: May 23, 2008

1. Introduction

"SANS 504: Hacker Techniques, Exploits and Incident Handling" illustrates the six steps to the incident handling process: preparation, identification, containment, eradication, recovery and lessons learned. This incident response system is derived from the SANS booklet, "Computer Security Incident Handling Step by Step: A Survival Guide for Computer Security Incident Handling". The two phases we want to take a look at in this paper are preparation and identification. While the other steps are important for the continuation of the business processes for your group, paying close attention during the preparation and identification phases can speed up your response time to an incident.

The preparation phase ensures your team has the skills and resources needed to respond to a potential incident at a moments notice (Skoudis & SANS, 2007). Some of the more common steps or concepts located in the preparation phase range from: policy, law enforcement notification (or not), peer notification, team oriented issues (team building, training, and organization), emergency communication, and finally the jump bag. A good security event management framework provides a suite of tools to assist you during the first four phases of the incident handling process. This paper focuses on the online tools that you can use that do not modify the suspected system, or "touch" it in an indirect manner so that we can use them without being concerned with the integrity of the data on the machine in case law enforcement needs to be involved. In the case that the tools would directly impact the machine for example, penetration testing software and some of the vulnerability scanners, the framework relies on data presented from past reports. What tools should be included in your “jump bag” kit for use during the identification phase? Which tools are essential when responding to an incident? We will look at these questions and more in depth in the subsequent chapters of this paper in order to provide a general roadmap that you can use when building a security event management framework for use in your business.

An incident is defined as an adverse event or the threat of an occurrence of such an event (also implied is the application of harm or intent to harm) to an information system or network (Skoudis & SANS, 2007). In the identification phase the incident response team gathers information, analyzes it, and determines whether an incident has occurred (Skoudis & SANS, 2007). There are so many tools that can be used to gather the needed information that it can be daunting to decide what is the correct tool for the situation or what information you actually need to gather for each particular incident. A good security event management framework can help make the identification phase more efficient by collecting the information pertaining to an incident in one location for easy analysis. This will help reduce potential financial losses due to extended downtime associated with the investigative process of incident handling.

In the first chapter we are going to take a look at the preparation phase of this project and take a look at some of the tools we want to integrate into our security event management framework. It is important to note that this framework will mostly be used for the systems internal to your business and that you should follow all policies laid down by your business before using some of the tools presented within this paper. For example, some policies may preclude the use of penetration testing software or may apply limitations to its use.

2. Framework Layout

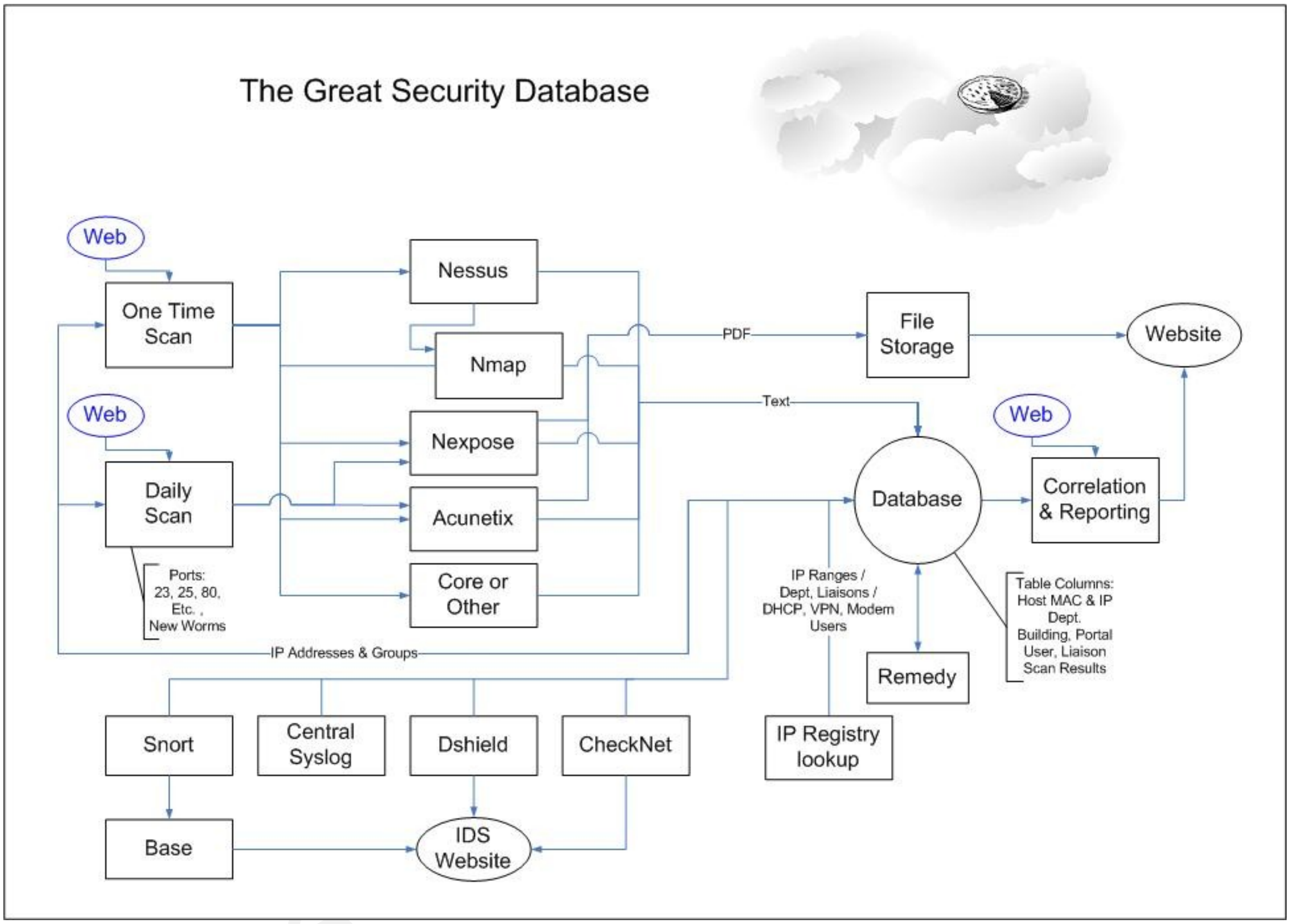


Figure 1: The Security Event Management Framework (Courtesy Philip Kobezak)

We will not go into the specifics of the actual hardware but will be working from the conceptual idea of what the hardware does as each organization will likely have their own requirements and specifications for what hardware can and should be used.

Figure 1 shows the database server that stores the event logs, reports, vulnerability scans, and other bits of information for analysis and correlation. The database engine that you choose to use on the server should have an acceptable data extraction and reporting mechanism.

The next item you will need is a file storage server that is capable of running a web server and has sufficient space to house the large files associated with vulnerability scans and event log reports. This information should not be confused with the reports sent to the database server. These reports will be in a non-text file format such as PDF, specifically, these are the high level reports for executive level personnel. The file server will also house the security review notes, reports and other associated files generated for each department or area reviewed. The web interface for this file server should not be public domain and should be limited to the incident handling team and other users as the need arises only, as it will contain the reports from various vulnerability scanners, penetration tests, ownership of systems, etc. Strict access control of this server will help ensure that this information is not exposed and then used to exploit your organizations infrastructure.

The first input set feeding into your database server are the vulnerability scanner and penetration testing data. The reports running to your database server should be text files that can be retrieved by the database as needed. The remaining formats (PDF, DOC, etc.) should be stored on your file storage system that will be linked to an internal website to assist you in your analysis.

The second set of inputs into your database is a listing of IP ranges that are within your organization and depending on how your organization is laid out you may wish to separate them based on department, with a listing of departmental liaisons. You should also consider using a breakdown of what type of connections as well, DHCP, modem users, etc, should the environment merit it.

The third set of inputs into the database will be the logs (event, host, router, firewall, etc) you collect at your organization. If you have a centralized IT structure you may have access to a central syslog server, if not you may have to rely on an alternate mechanism to gather as many logs as possible. This could hamper your efforts a little bit, but most decentralized groups will respond favorably when you explain that you are working to better secure their information as well as the organizations. One collection method you could employ is DShield or perhaps a home grown solution of your own. We will look at DShield a little more in depth in the tools chapter.

The final item attached to the framework is a trouble ticket tracking system that will communicate with the database to speed responses to those parties involved, track steps involved in the incident, etc. This should not be your only means of keeping track of how the incident is unfolding, as it can be subject to compromise, corruption, and crashing just as any computer system is. Sometimes you just can’t beat the tried and true method of a notebook and pen.

3. Preparation – Vulnerability Scanning and Penetration Testing Tools

One of the first questions that you should consider when building your framework is what combination of commercial vs. freeware software you want to run. Usually the biggest factor in this question is what can your business afford? Another factor to keep in mind is your organizations stance on freeware, some dot com businesses do not support freeware and may have rules or policies prohibiting their use within the organization. However, many of the freeware products can be just as useful as commercial products. For example, we will be looking at a few commercial products that we currently use at Virginia Tech.

The first thing to keep in mind with this framework is that you are going to want to run more than one vulnerability scanner. Correlation between scanners can help eliminate false positives and start to narrow down what type of incident may have occurred. Vulnerability scanners should have the following features if they are integrated into a security event management framework. First, you want a scanner that produces both readable and useable reports based on the results of the scan. Once an incident is under way the handler may be called upon to present initial reports to the different levels of management involved in the company. You want a set of reports that you can hand off to the most technically inexperienced person so that they at least have an idea of what is going on. Second, you want a scanner that offers a good selection of vulnerabilities that it will scan for. This can commonly be found by how many "Common Vulnerability and Exposures" (CVE) the product advertises that it scans for. This will increase the time that the scan will take, but can provide you with some valuable insight into the type of incident that may have occurred. Using a scanner that looks for a larger number of vulnerabilities will force the handler to look at many different vectors instead of becoming target locked. Having a focused scanner, on the other hand, can also be a good thing, as it will likely scan faster and give results quickly as a first pass in order to determine how bad the incident may be.

Along with the vulnerability scanner we want to include at least one penetration testing software. We want to make certain that we can eliminate false positives within our system as well as take a look for other vulnerabilities that may have gotten glossed over with a generic vulnerability scanner. As always, please note that you should alert your users and ask permission (as required) to conduct penetration tests on systems within your business as they can be highly disruptive.

3.1. Vulnerability Scanning Tools - Freeware

Nessus is a good choice to start with in the realm of vulnerability scanners. Nessus has both a freeware version that you can download and run out of the box and a paid version that will give you a few more features that you may find useful in your environment. The free version of Nessus is more than capable of serving as a good vulnerability scanner. A focused vulnerability scanner for certain situations provides some correlation/redundancy as well. One area that we have chosen to focus on is web applications. The security vulnerabilities in these applications allows for disclosure, deletion, and/or changing of information from a database and the information they have access to lends them to be very tempting targets. Acunetix, a web application vulnerability scanner, provides both a free version and a paid version as well. The main difference here, however, is that the free version only scans for cross site scripting (XSS) vulnerabilities, while the paid version scans for SQL injection, JavaScript coding issues, XSS, and more. It may behoove your organization to invest in this tool on a paid level, especially if you use, maintain, and/or develop web applications.

While not exactly a vulnerability scanner, we did choose to include Nmap into this group as well, Nmap being a port scanning tool that will alert you to any and all open ports on a given machine. Nmap is not limited to just providing a port scan on a given system as it is also capable of returning additional information such as operating system and version, MAC address, reverse DNS names, etc.

3.2. Vulnerability Scanning Tools - Commercial

One of the commercial vulnerability scanners that our organization uses is the commercial product Nexpose, from Rapid7. We have found this to be a significantly feature rich scanner as it scans for web app vulnerabilities, database vulnerabilities, network vulnerabilities and more. Another factor, as mentioned previously, that we took into consideration was the reporting mechanism of Nexpose. The reports generated from this scanner are very well laid out and easy to understand no matter your technical level. They provide both a well laid out executive summary as well as an in-depth technical summary of all vulnerabilities found from the scan.

Acunetix, as mentioned previously, is another commercial product Virginia Tech uses for certain situations and events. Acunetix is a more specific scanner that focuses on web application security and the vulnerabilities associated. Running both Acunetix and Nexpose against a target machine helps with correlation for the potential incident, assuming that both scanners agree with the vulnerability assessed.

3.3. Penetration Testing Tools – Freeware

Metasploit is the freeware penetration testing software we have chosen to integrate into our security event management framework. Metasploit is a widely known and supported open source project available at http://www.metasploit.org/, "The Metasploit Framework is a development platform for creating security tools and exploits." ("The Metasploit Framework", 2003-2008) Specifically, the incident handler can create modules within the framework to run against a given system to exploit either a known or discovered vulnerability in order to compromise the system. Once the exploit has been confirmed via distribution of a payload on the target machine it is up to the system administration to take the appropriate steps to close off that particular vulnerability. Many penetration schemes work hand-in-hand with vulnerability scanners to verify that an exploit actually does exist and has not been reported as a false-positive. The Metasploit Framework has the advantage over most products, including commercial options, in that the modules can be specifically written for the parameters within your organization and changed on the fly. This flexibility can allow for the modules to be tailored so they do not exploit systems in a destructive manner so that the penetration test can be run on a more frequent schedule.

3.4. Penetration Testing Tools - Commercial

In this example we have included Core Impact as our penetration testing software. Core Impact is a commercial product that contains a suite of pre-defined exploits that can be run against your systems. Some of the exploits defined within the Core Impact framework are destructive in nature and can cause loss of system performance or denial of service if you choose to run them. As previously mentioned you should always alert your users to any penetration testing you plan to do and fully disclose the risks involved. Our goal is to protect the machines and data, not to cause a loss of business within our own organization. Metasploit can be targeted to one set of exploits dependant on the module that is written and run. Core Impact attempts to provide the end user with an entire arsenal of potential attacks to run against the machine. So with penetration testing software we also have the broad range tool versus the targeted or specific tool, each having their place in a security event management framework.

4. Preparation – Tools – Dshield and other log gathering

Along with the active tools: vulnerability scanners, penetration testing, command line (ping, traceroute), etc. you also need to make certain that during an incident or preparing for one that you have good log collection and analysis tools. Sometimes the only information you can get about the vector of attack can be found in the logs of the affected machines, thus it is imperative that you maintain at least some logs within your organization. This, of course, is not a small undertaking as the amount of logs generated, even from small organizations, can be immense. It is highly recommended that you make certain that you have the appropriate storage capacity in order to maintain at least seven days worth of logs that are readily accessible. Longer periods of time will increase the amount of space required, but will also increase the likelihood that you will be able to track down the initial point of attack from the incident and any other relevant pieces of information that go along with it, phone home capabilities, secondary vectors, etc.

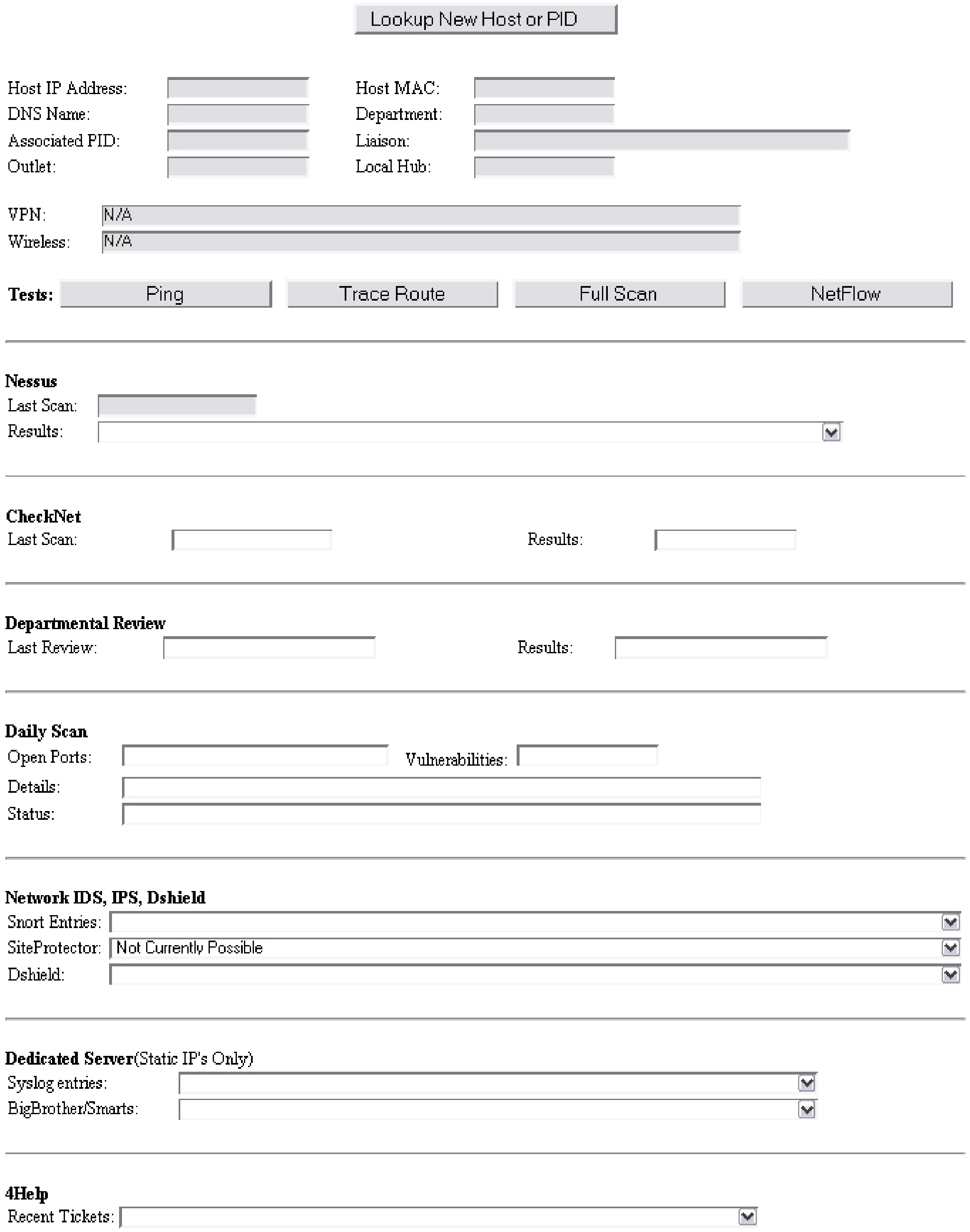
Multiple sources and types of logs are highly recommended so that correlation of events is easier to spot. Some of this can be made easier if your organization has a central IT structure that allows for a central syslog server. If your organization does not follow the central IT structure, there are some options open to you to help you collect the logs you will need. The foremost in these, and a solution that Virginia Tech uses, is the DShield. The DShield goes a little beyond a straight syslog function in that it is a distributed intrusion detection system that takes firewall logs from various sources and attempts to combine them for you. An example of DShield at work in a "local" setting can be seen at http://dshield.cirt.vt.edu/ while a "global" view can be seen at http://isc.sans.org/. The one thing to note about the DShield is that it is only as effective as the number of logs that are sent to it. Currently this is a manual adoption policy so you will need to make certain each department in your organization is on board with this project. Once you have a good record of logs being fed to the DShield you can start to look beyond the “reactionary” uses of DShield and start looking at trending and truly preparing for an incident. The DShield will give you a good idea of what is being probed for a potential attack.

Snort is a good freeware solution as a software driven IDS and is integrated into our framework as show in Figure 1. Snort will likely provide you with more detailed and specific information regarding the potential incident as Snort uses a rule based approach that provides more granularity in the rules used. A firewall rule-set provides a generic set of rules to follow for monitoring, while Snort and other IDS/IPS solutions define rules based upon known attack vectors. There are commercial solutions for IDS/IPS as well, both in a hardware configuration and software configuration depending on what your organization needs. In order to make the best use of your IDS solution the sensors will need to be placed in the appropriate locations in order to grant the most access to network traffic. This will give you plenty of logs to go through in the event of an incident. It should be noted that an IDS/IPS should not be the only solution as there are a number of viruses, attacks, etc. that are specifically written to get around an IDS/IPS solution. However, an IDS/IPS helps to guarantee that there is more than one solution to look at so that chances of detecting an incident prior to a compromise are higher. Just as attacks take a multi-vector approach your defense-in-depth should follow the same suit and you should take a multi-vector approach to information gathering. In the next chapter we will take a look at how this framework will function once we have it built.

5. Identification – How does the framework work?

Based upon the steps and ideas presented in the previous sections you should have the preparation phase done for your framework project and have it built. Since there are a fairly large number of tools that are being integrated with this solution it is imperative that you fully test the framework once you have it built so that you are not surprised once an incident occurs. This can also be a benefit because you can use this testing time as training for your incident handlers as well, giving them a scenario to follow that they will need to use the framework in order to resolve. It would be a good idea to use an incident scenario that you are all familiar with so that you can expect specific results from the various areas the framework is reporting on. This will give your testers a script to follow, of a sort, and expected results to compare the tests against.

Let’s assume that a single machine within your organization has been compromised and you need to begin the identification phase to determine the extent of any data loss and then move on to the containment phase. Best practices probably indicate that you will need to search for any vulnerability scans you have performed in the past on this machine, whether there were any scanning attacks from an outside source on this machine previously to determine if there are open ports, if there were any trouble tickets created by this machine’s operator for mysterious behavior, etc. As we are all familiar with, this part of the incident handling process can take a significant amount of time as we gather all of this information together, but as we see from the image below some of this time can be cut off through our framework.



Host IP Address:
DNS Name:
Associated PID:
Outlet:

Host MAC:
Department:
Liaison:
Local Hub:

VPN: N/A
Wireless: N/A

Nessus
Last Scan:
Results:

CheckNet
Last Scan:
Results:

Departmental Review
Last Review:
Results:

Daily Scan
Open Ports:
Vulnerabilities:
Details:
Status:

Network IDS, IPS, Dshield
Snort Entries:
SiteProtector: Not Currently Possible
Dshield:

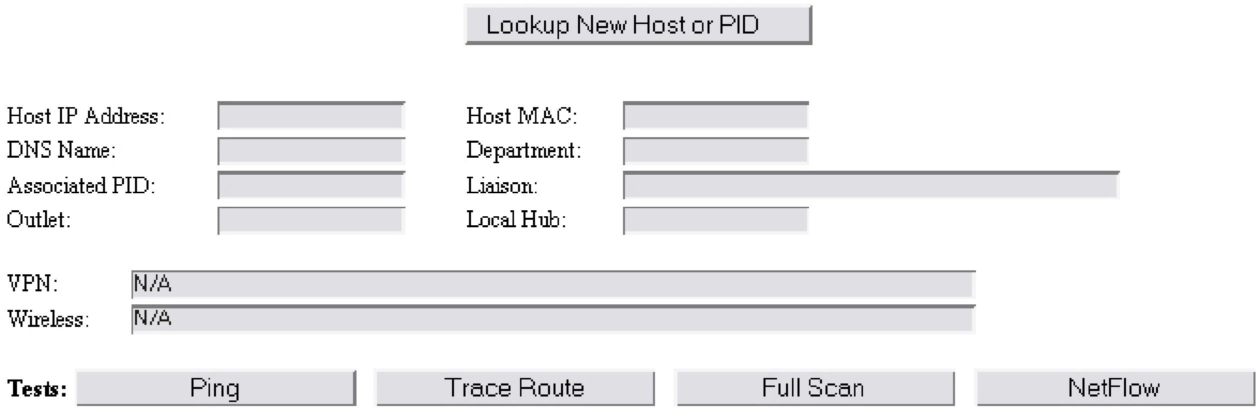
Dedicated Server(Static IP's Only)
Syslog entries:
BigBrother/Smarts:

4Help
Recent Tickets:

Figure 2: The Security Event Management Framework web interface *(Courtesy Philip Kobezak)*

Figure 2 shows the end result of all the work done in the preparation phase. Notice that each of the tools mentioned in the previous sections are represented in this web interface and consider how this layout can speed up response time to an incident. This interface is the model that Virginia Tech will use and it is still under development to include all of the features we think we will need. If there is a difference in what the current model looks like and what we want the end product to look like we will point that out in the description of the specific areas of the web interface.

5.1. How does the framework work? – System Information



Host IP Address:

Host MAC:

DNS Name:

Department:

Associated PID:

Liaison:

Outlet:

Local Hub:

VPN: N/A

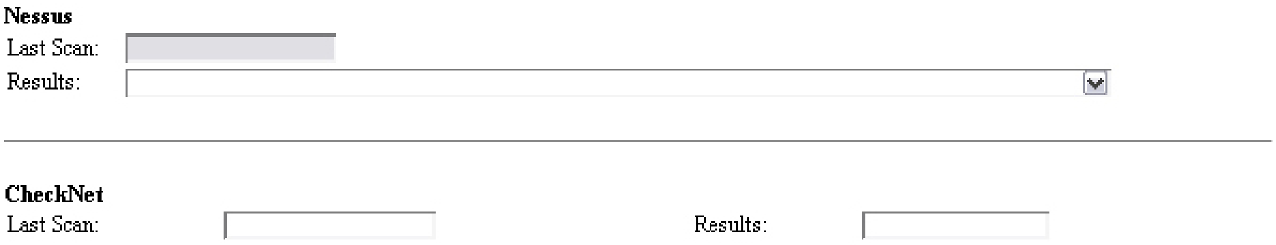
Wireless: N/A

Figure 2a: S.E.M Web Interface – System Information *(Courtesy Philip Kobezak)*

Figure 2a gives us the system information on the particular machine we are looking up. To start the process click the "Lookup New Host or PID" button and provide either the IP address or the Personal Identifying Information (PID) for the compromised machine. Once the IP address or PID is entered the interface will return the following information where: the machine is located, MAC address, liaison (system admin) for the machine, and what wall-jack it is located on (should there be more than one computer connected in one area). The DNS name, MAC address, associated PID, Department is all normally returned by the result of the liaison field, which is also included. Each department or group within Virginia Tech has a system administrator or other responsible person that serves as the communications departmental liaison. They are responsible for maintaining a list of their equipment, their IP address range (and what machine is connected to which IP address), and which outlet (wall-jack) for each machine. Our communications and network services group such as returns some of the information, the local hub the machine connects through, if it connects to/through a VPN, and/or if it uses the wireless system to connect to the network.

In addition to the provided information several manual tests can be run to check the integrity of the machine. Ping, to check if the machine is reachable, traceroute to determine where a break in connection may be occurring if the machine is not reachable, full scan which currently runs a Nessus scan against the machine you have looked up, and NetFlow which displays all of the packet traffic for the given machine. Virginia Tech is looking at changing the full scan option to relate to each of the different vulnerability scanners used at the University so that each test can be run independently with the scanner of choice. NetFlow is the packet traffic captured, both inbound and outbound. Clicking the NetFlow button will display a history of packet traffic for the machine, showing spikes in packet transfers but no other data. This can be used as another means to detect if the machine may have been compromised as the packets during certain time frames that don't see "normal" use may be larger than expected or there may be an even larger increase in traffic than normal during a regular day. While this data is not very descriptive it can help with trending a machines network usage so you have a good idea what the baseline for the machine is. Once we have completed the work on this portion of the framework we should be able to pull the packets for the day, week, month, or year depending on how much data is stored (each file is at least 5 gigs). Once the data has been pulled we will have a graph for each of those time periods that will display day/week/month/year vs. time of day so there is a visual representation for a quick look with the actual values located underneath the graph in a table.

5.2. How does the framework work? – Vulnerability Scanners



Nessus

Last Scan:

Results:

CheckNet

Last Scan:

Results:

Figure 2b: S.E.M Web Interface – Vulnerability Scanners (Courtesy Philip Kobezak)

Figure 2b provides access to the past vulnerability scanner reports that were run on the machine being viewed and provide the incident response team with a view of any anomalies or reports that suggest there may be a vulnerability located in the machine. Currently we do not have any functionality built in to do a scan on the IP address itself beyond a limited Nessus scan using the FullScan option, building that functionality in is the next logical progression. Looking at past histories can be helpful, but should not be completely relied upon as any number of things could have changed the configuration of the machine between time of last scan and time of potential incident. The current plan is to develop and implement a "Run Scan" button next to each of the vulnerability scanners we have connected to the framework so there is no question as to which scanner is running and what type of results we should expect. The results drop-down will link to the report for the last scan of the machine and list whether it passed or failed so that we have a good idea of what to expect before we click on the results to see the details. CheckNet is also related to Nessus since it uses Nessus with specific rules defined for a quick scan to check certain ports and vulnerabilities that are common in our environment. CheckNet was an automated scan performed on each machine the first time it connected to the Virginia Tech network to ensure that the machine met the minimum security guidelines to ensure network and user safety. Machines that passed the initial CheckNet scan were given full access to the Virginia Tech network, while machines that did not pass were only allowed on a quarantined network. This option has been disabled due to a lack of hardware needed to maintain the scalability of the CheckNet system during certain periods of increased activity (beginning of a new semester, football games, etc.).

5.3. <u>How does the framework work? – Departmental Review</u>

Departmental Review

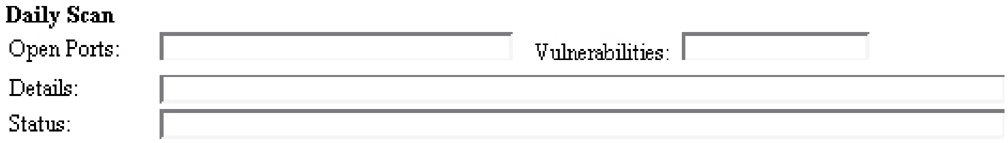
Last Review:

Results:

Figure 2c: S.E.M Web Interface – Departmental Review (Courtesy Philip Kobezak)

Figure 2c returns the date of the last departmental review and the results associated with the review. A departmental review consists of various scans (vulnerability, find SSNs and CCNs, etc.) as well as a penetration test to determine the security of the systems located within the department. If we should find that this particular system was involved in a review recently we can also see if they had any issues that needed to be fixed. This also may provide insight into the potential attack vector and should provide some insight into what type of information may be involved, confidential or not. Should the date of the last review not be recent this information may not be entirely relevant to the situation, but it can provide some background information regarding the general practices surrounding the machine's maintenance, etc.

5.4. How does the framework work? – Daily Scans

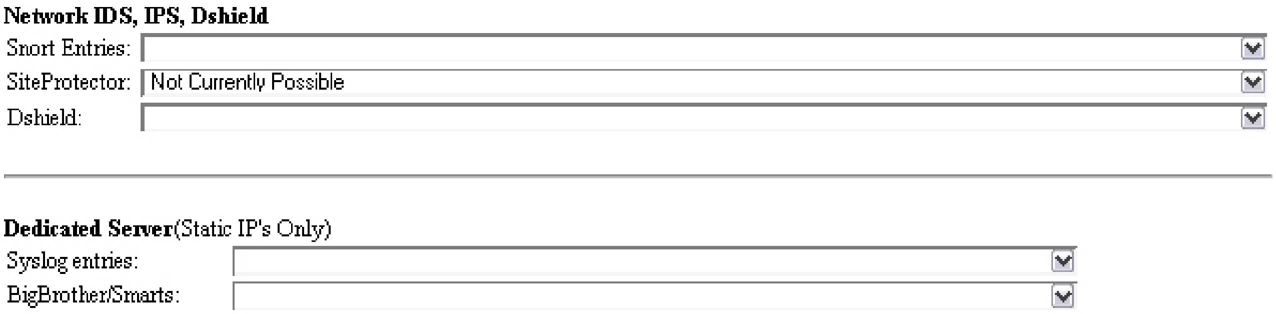


Daily Scan
Open Ports: Vulnerabilities:
Details:
Status:

Figure 2d: S.E.M Web Interface – Daily Scans (Courtesy Philip Kobezak)

Assuming you have the bandwidth, and as the case may be, a suitable number of free "seats" in a commercial product, the daily scan section would be a vulnerability scan set up to run each day against all of the machines in your organization with limited rules/vulnerabilities to look for. This may not be feasible in a large organization so you may have to use a more lightweight tool to just scan for open ports on the machine. Another useful section to add here, as illustrated in the Daily Scan section would be to tie the reports together each day with quick summary that states "Unchanged as of 'X date'" or alerts you to any changes that may have taken place.

5.5. How does the framework work? – Logs: Dshield, Snort, etc.



Network IDS, IPS, Dshield

Snort Entries:

SiteProtector: Not Currently Possible

Dshield:

Dedicated Server(Static IP's Only)

Syslog entries:

BigBrother/Smarts:

Figure 2e: S.E.M Web Interface – Log Gathering (Courtesy Philip Kobezak)

Once you have exhausted the vulnerability scanner options we then move towards some of the log gathering mechanics we have tied into our framework, starting with Snort and DShield logs. Has this particular IP address been targeted either as a victim in the Snort logs or as a machine attacking an outside source in the DShield logs? Keep in mind that the potential compromise may be an elaborate “man in the middle” attack using one of your machines as a relay point to get elsewhere. Looking at the outgoing connections from a particular IP address is more important then looking at the inbound connections and requests since this tells us how “clean” your network is. Coupled along with the Snort and DShield entries are the standard central syslogs for the machines as well. Should the machine in question be a server that is connected to a BigBrother, system monitoring for up time, software than you may find an entry on whether the machine is currently up or down.

5.6. How does the framework work? – Trouble Ticket System

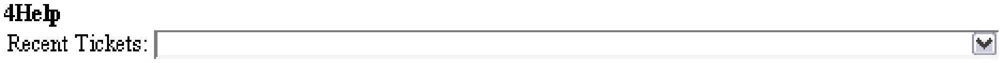


Figure 2f: S.E.M Web Interface – Trouble Ticket System *(Courtesy Philip Kobezak)*

And the final portion of this form is the connection to the problem reporting/tracking system used in your organization. Has the machine had a trouble ticket created recently, did the user report abnormal behavior and/or error messages? What steps were taken to attempt to resolve these issues and were they helpful? In the completed version of this part of the framework there should also be an option to create a trouble ticket if it has not been created, as another means to track the progress of this incident. It is entirely possible that you may not wish to tie the trouble ticket creation to the "main" helpdesk environment so you do not cause any confusion. This is illustrated if one of the people staffing the helpdesk did not read the ticket completely before attempting to contact the user and walk them through troubleshooting steps thereby potentially invalidating any evidence collecting that may need to be done due to the nature of the incident. An internal reporting mechanism that is used specifically for tracking incidents and the steps used in the investigation process may better serve your organization in this case.

6. Benefits

The time it takes for the incident to be discovered, investigated and then resolved is directly proportionate to the amount of money or data your organization could be losing. The security event management framework detailed in this paper will help make the incident handling process more efficient by speeding up some of the more time consuming tasks involved with gathering the needed information to respond to an incident. This framework allows an investigator to retrieve archived copies of vulnerability scans and penetration tests from databases in a timely fashion, this helps reduce the amount of time spent in identifying and containing an attack. The integrated web site will speed the process further along by allowing access to most of the tools that you need from multiple locations at a moments notice, especially useful if you do not have your full jump kit at your immediate disposal.

Organization of the notes, data, steps taken, etc. is another critical factor in the incident handling process, not only from a "solution" standpoint but from a legal one as well. If the incident needed to be referred to a legal authority all of the notes, steps taken, and information gathered would need to be beyond reproach in the instance that a case was brought before a court. While the framework does not guarantee that everything will be in order, it does provide guidelines and applications to assist the investigator in the organization of the data involved in the incident. All too often important pieces of information can be lost in the cracks or overlooked when in the midst of an incident. The security event management framework can help maintain the organization of the incident by providing a step-by-step guide to walk you through the identification process, with each step's information recorded within the framework. The framework will allow the incident handler to take each step in stride and hopefully relieve some of the stress associated with an incident so they can further concentrate on maintaining a thorough and complete investigation of the incident.

Making use of the framework can also assist in the "lessons learned" phase of the incident handling process as well. This benefit can be two-fold by providing the incident handler with the information regarding the incident, but also providing valuable information to make changes/upgrades to the framework. The same convenience gained from having all of the tools together in one location during the identification phase is also applicable during "lessons learned" by storing all of the information regarding an incident in one location. Looking beyond the "lessons learned" phase and circling back into the preparation phase for the next incident, you can also use this information to develop scenarios to test against the framework and your team. Each incident or scenario that you run through the framework should also show you any areas that need to be improved. Just as each attacker evolves a set of tools to get around the system, the security professionals are also evolving tools to assist in their jobs. You should never be happy with keeping your framework static and should always look for things to change or add to it to make it stronger.

7. Conclusion

Now that we have looked at the underlying mechanics associated with a security event management framework it becomes apparent how useful something like this can be. Populating much of the information needed in a potential incident with a single click just with the PID or IP address associated with the potentially compromised machine. Instead of spending time attempting to gather the basic information of the machines involved in the incident, the handler can spend more time in the containment and eradication phases of the incident handling process. With a little more time spent in preparation we find that the identification phase can be streamlined to provide only the information needed in a potential incident. In the perfect world a tool like this would not be needed as no one would be attempting harm someone else's IT assets, but since we do not live in this world anything we can do to make our jobs easier and more efficient is a good idea.

8. References

Skoudis, Ed, & SANS (2007). “Security 504: Hacker Techniques, Exploits, and Incident Handling”. 5-46

Metasploit LLC. (2003-2008). “The Metasploit Framework”. Retrieved April 30, 2008, from http://www.metasploit.org/framework/

Northcutt, Stephen, Alexander, Bryce, Anderson, S., Balodimos, Connie, et al. (1998). “Computer Security Incident Handling Step by Step: A Survival Guide for Computer Security Incident Handling”

Kobezak, Philip. Virginia Tech Information Technology Security Office, Unpublished Technical Report.