Global Information Assurance Certification Paper

Implementing Real Secure and Internet Scanner: A Case Study

Abstract
Unfortunately, my company is among the countless others whose management's attitude on information systems security was a hands-off approach: "It won't happen to us". Thankfully, that is slowly changing with the increase of the topic in the media. Until recently, our only security component was a firewall, which is only the first layer of implementing a "Defense in Depth" security strategy. I now have implemented an Intrusion Detection System (IDS) and a vulnerability assessment tool to add another dimension to our security infrastructure.

This is a case study of the steps I have taken to implement both products, The Internet Scanner version 6.2.1 and RealSecure version 6.6. Both products are from Internet Security Systems (ISS). This paper will include an overview of the security measures in place at my company prior to the installation of the products, an overview of how the products work, examples of how the software was implemented and tested, and finally a discussion of how the products enhanced the information systems security infrastructure at my company.

Before

A brief overview of the current company network infrastructure will help to give an idea of what I have to protect. I work in one engineering division of a worldwide automobile parts manufacturer. At my location there are network connections directly to manufacturing plants in the US and Mexico. There is also a connection to our headquarters in Germany. Headquarters is connected to many other locations all over the globe. We have a T1 connection to ANX (Automotive Network Exchange), which also provides access to the Internet. There is a firewall between our internal network and this ANX/Internet connection. There is a DMZ configured off of the firewall where an FTP server resides. It is currently the only item in the DMZ. Our local area network is comprised of Novell, NT and Unix servers, with approximately 300 PC's and a handful of Unix workstations. As shown in Figure 1, there is no firewall protecting the local area network from the other corporate locations. Corporate Policy dictates that there must be a firewall protecting the company from any connection to the Internet, but we are not to place a firewall between our network and the other divisions. This not only means we have to protect ourselves from any malicious attempt from within our own division, but also on a corporate level, of 21,000 employees. The 2002 Computer Security Institute/FBI Computer Crime and Security Survey results indicate that 74% of the respondents cited their Internet connection as a frequent point of attack while 33% stated their internal systems were the attack target (Power, p.4). 75% stated that disgruntled employees were the likely source of the attack (Power, p.8). With 21,000 employees, soon to be 50,000 after a merger, we need more security measures in place than just one firewall on the perimeter.

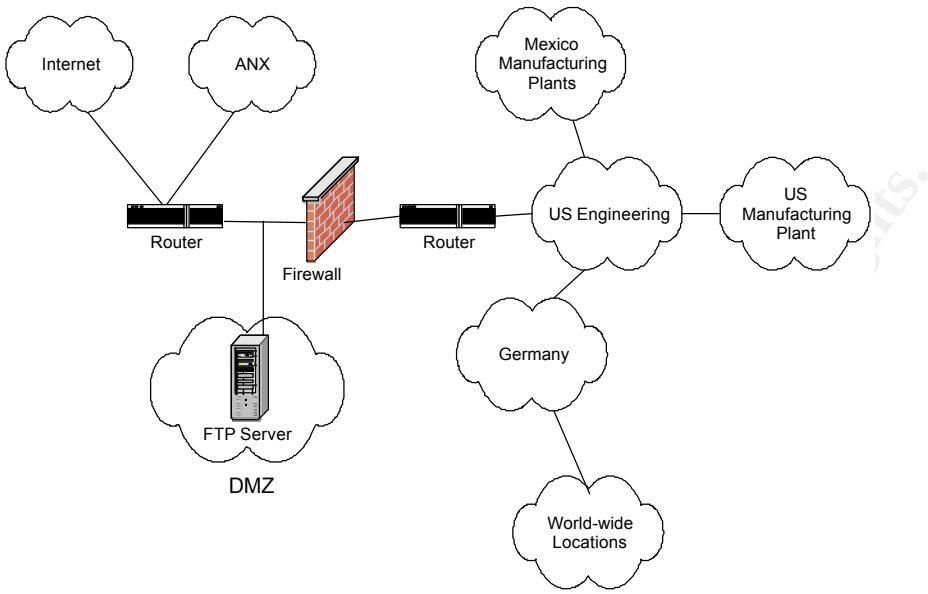

Figure 1.

Product Summary

The vulnerability scanner, Internet Scanner, and the IDS, RealSecure, are from Internet Security Systems (ISS), an Atlanta, Georgia based company, founded in 1994. These products were chosen based on the following considerations: ISS's large install base of over 9,000 corporations including 49 of the Fortune 50, the favorable reviews found on the Internet, and the recommendation of our contract support provider. Since our LAN is a switched network environment, we did not purchase any network sensors. To attempt to use a network sensor in a switched environment we could configure a spanning port or place one on a network hub or tap. Although, even with one of these options, network based IDS can drop packets in a high-speed environment. This will be a future consideration, after I become familiar with the use of Internet Scanner and RealSecure, as well as researching the best way to implement a network sensor in our switched LAN.

The RealSecure product is a signature based IDS. A signature based IDS is one which "looks for activity that matches a predefined set of events that uniquely describe a known attack". (ITL Bulletin Nov 99). RealSecure server sensors protect a server via packet interception to analyze: 1) the network traffic to and from the server, 2) the operating system log entries and 3) kernel-level events. It compares this data with the signatures known attacks. Signature based IDSs are good for detecting attacks without generating many false alarms, but that also means that they must be constantly updated with new attack signatures. ISS provides a feature to update the software, called X-Press Updates, or XPUs. XPUs are software releases, which includes a DLL or shared library that contains new security content and new or revised signatures and checks. Users of the products can join a mailing list for automatic notification when new XPUs are available to download from the ISS website.

Installation

The initial purchase of RealSecure was comprised of two server sensors. I have installed the server sensors on our FTP server in the DMZ and the internal DNS server. Both servers are Windows 2000 running IIS. I chose to install the sensors with the network monitoring option. Even though this will only monitor the network activity to and from the servers, it will hopefully give me a good idea as to what kind of network activity is directed at the firewall as well as probing my internal network.

The RealSecure sensors are monitored and configured via the Workgroup Manager. The WorkGroup Manager has three components: a console, event collector and database. Depending on the size of the deployment, each component could be installed on a different computer or for a small installation of one to five sensors, as in my case, all components can be installed on one machine. I have installed the WorkGroup Manager and all components on an Intel Pentium PIII 550 MHz PC, with 264 MB RAM. The console controls all of the sensors, displays the alerts, and generates reports. The event collector collects the data from each sensor and sends it to the console and the database. The database stores the sensor data collected by the event collector and the information about each sensor and the event collector. The default database is Microsoft Data Engine (MSDE), which I am using. MSDE has size limitation of 2GB of data. For installations where this is not enough, RealSecure can also be configured to use a SQL database.

The actual installation of the software is your typical point and click, but it is helpful to understand how the components communicate to ensure a successful implementation. The communication between the RealSecure components occurs in real-time on an encrypted channel. This is to prevent knowledgeable intruders from intercepting the IDS traffic. The secure channel is created using authentication to positively identify the components to each other. During the installation a public and private key are generated. There are two cryptographic providers installed by default, Certicom and RSA. Certicom is listed as ISS ECNRA and RSA is listed as Microsoft Enhanced. In order for the sensors and the event collector to communicate with the console over a secured channel, they must each have a copy of the console's public authentication key. Likewise, for the sensors to pass data to the event collector they must also have a copy of the event collector's public keys. The correct placement of these keys is imperative for a successful installation. ISS has provided an auto-import feature to automatically copy the public keys upon the initial communication between each component. I used this auto-import feature and had no problem with the installation. Although, this could have been a security risk if the sensor first received a connection from an unknown user, therefore preventing the sensors from communicating with my console and event collector.

There is an ISS daemon running on each machine that contains a server sensor

or event collector. The daemons act as intermediaries between the sensors or event collector and the console. The console communicates with the daemons that then pass the commands on to the sensor or event collector. The daemon then responds back to the console with the status of the command. The ports used for this communication can be modified, but I have used the defaults. The console uses any port available on the system. The ISS daemons listen on port 2998. The server sensors listen on port 902. The event collector listens on port 903. If the sensor lies on the opposite side of a firewall than the event collector and console, the firewall must be configured to allow this traffic to pass. Since I have installed a sensor on the FTP server in the DMZ, I have defined a rule to allow traffic on these ports to pass specifically between the Workgroup Manager and the FTP server.

For Windows installations the Server Sensor is integrated with a BlackICE agent. This agent has two components. The first is a packet capture module that examines the entire packet as it passes the NIC and NDIS driver to see if it matches any signatures. The second is a firewall component that inspects only the packet header. If a security event is not detected, the packet continues up the IP stack. If one is detected, the event is reported as an alert to the server sensor.

Once the WorkGroup Manager components and the server sensors are installed all management and monitoring of the sensors and the event collector is done via the console. The console screen is separated into five windows. Three windows display the real-time events broken down into high, medium, and low priorities, each with its own window. ISS defines priorities as follows: high priority alerts are those that allow unauthorized access to the server, medium provide access to sensitive network data that may lead to a high risk exploit, and low are those that allow access to sensitive data but will probably not lead to a higher exploit. On the left side of the console is the Activity Tree window that also displays the events, but grouped by source, destination or the event itself. The fifth window across the bottom is the Manage Assets window. ISS assets are the daemons, sensors, and event collectors. In my deployment I have five assets: two server sensors, two server sensor daemons and one event collector daemon.

I found that each server sensor has a myriad of options available to configure, for each signature. The server sensor comes with ten predefined policies to help in determining which signature is appropriate for the server it is installed on. The more signatures that are turned on the higher the RealSecure resource utilization on the server. ISS states that a well-defined policy applied to a server sensor will have a 6-8% performance degradation on the server. Each predefined policy is configured for a specific type of operating system, Windows, Linux, or Solaris and whether or not it is running as a web server. These policies can and should be modified to fit the particular server it is on and to conform to any corporate requirements. Modifying the policies involves

configuring and creating signatures. RealSecure categorizes the signatures into four groups. The first is the Protect group, which focuses on intrusion prevention rather than detection. The Protect signatures work like a firewall by monitoring and responding to specific network traffic that meets certain criteria, such as allowing only local subnet traffic to access the server. These are only for inbound traffic to the server. The response could just be an alert or could actually block the packets. The protect signatures also look for suspicious port and service activity. The second signature category monitors network traffic and triggers an event when the network activity matches the predefined criteria. The Network Events are grouped into 21 categories such as: Back Doors, Denial of Service, DNS, FTP, HTTP, ICMP, IP, POP, RPC, Scanners, SNMP, etc. Each category then has multiple signatures. For example, at the time of my installation, the HTTP group contained 57 different signatures. The third signature category is an OS Event. These events watch the activity that occurs in the system log files. RealSecure breaks these OS events down into groups by the operating system in which the sensor is installed on and a group that applies to all operating systems. When modifying a policy, you turn on or off items that you want to monitor and respond to, as well as set specific properties of the event. Each event has up to six response types: display the event to the console, log the event to the database, send an e-mail notification, reply to the intruder with a banner such as “No Trespassing”, set an SNMP trap, or perform a predefined action based on secure logic. You can choose to use one or all response types. The policy I installed is based on the out of the box “Maximum_Windows_IIS” policy.

Installing the Internet Scanner software was also a very simple, standard software install, that can be put on a low end computer. I have installed it on a PIII 450MHz Intel PC with 264 MB of RAM. Again, once installed determining what machines and what vulnerabilities to scan for was the hard part. As with RealSecure, Internet Scanner can be updated with the latest vulnerability checks with the X-Press Update feature. At the time of installation, after I installed the latest XPU, the exploit list contained 1145 exploits. To help determine which exploits to scan for, Internet Scanner has grouped them into five levels of policies. A policy is the group of exploits scanned for during one session. The more vulnerabilities checked for during a particular session, the longer the scan will take. According to the ISS documentation the levels are as follows: Level 1 will take an inventory of the scanned machines to identify the type of operating system, a Level 2 policy identifies the services running, level 3 looks for holes that can be compromised by unskilled attackers and signs that the system has already been compromised, level 4 checks for vulnerabilities that can be found by automated attack tools, and a Level 5 policy looks for vulnerabilities that highly skilled attackers would exploit and system misconfigurations. As I will point out later, the L1_Inventory scan also determined what services were running in addition to other checks. ISS documentation recommends that the majority of systems should be scanned by a Level 3 policy. A Level 5 policy is for systems in a high risk location, like a

DMZ. Each predefined policy can be used as is, or used a template to configure a new policy relevant to the organizations needs. Depending on the vulnerability checked for, each exploit may have many options to customize, such as which port or ports to run the scan on.

Testing

My initial scan using the Internet Scanner was of the FTP server and the DNS server. This would help me verify how well the server sensors detected activity. I first ran the predefined L1_Inventory scan. I did not modify the scan settings from the default as provided by ISS. This scan successfully determined: the operating system, DNS Name, NetBios name, NetBios domain, and the services running on each open port, for the DNS server, and only the OS and services for the FTP server. In addition, this scan produced many alerts at the RealSecure console, as it should have. The alerts are displayed in the Activity Tree in the upper left window of Figure 2.

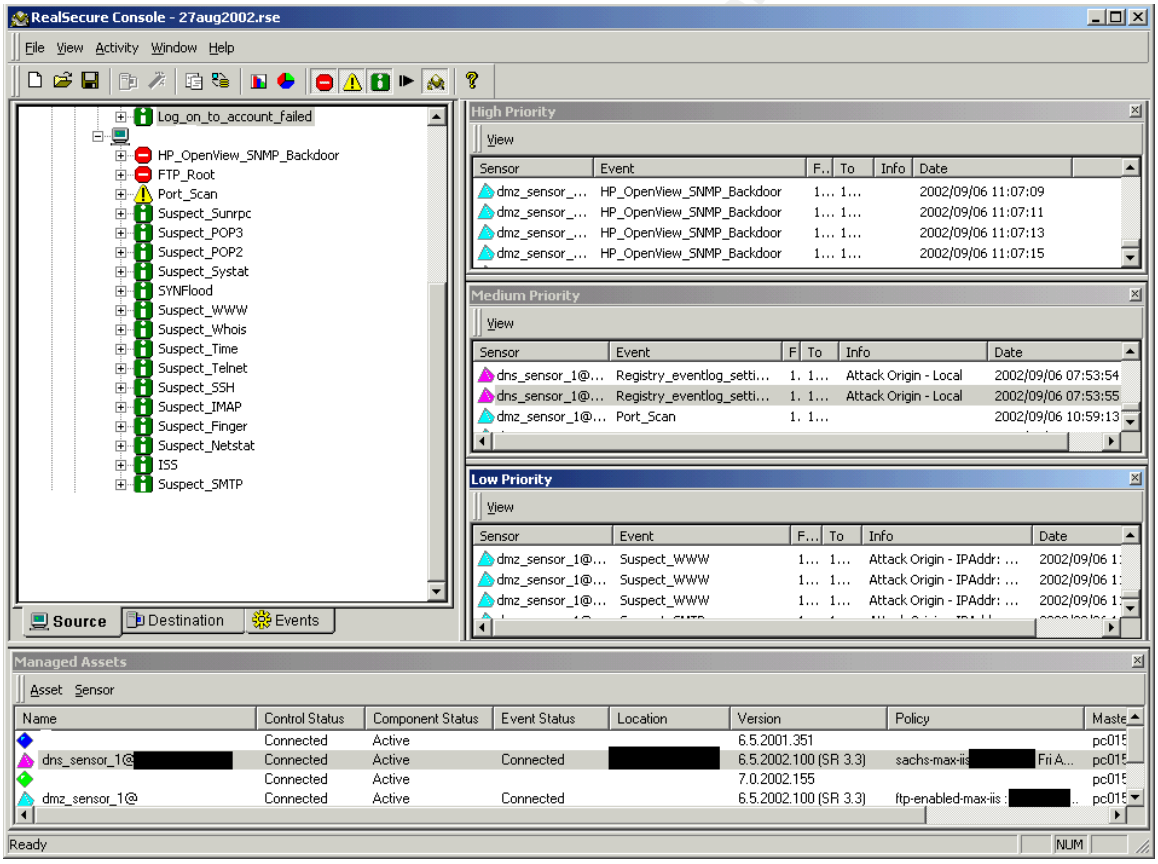

Figure 2.

All of the IP addresses have been blacked out from the above and following figures. The top two events in red are classified as high priority, the port scan is medium, and the rest are low. When comparing these to the actual exploits scanned for by the L1_Inventory scan I found that the HP_OpenView_SNMP_Backdoor alert was triggered by a SNMPv2Discovery or

SNMPv1Discovery check in the L1_Inventory scan. These checks are used to determine if SNMP is running. When inspecting the event from the RealSecure console, it logged this alert 59 times. See Figure 3.

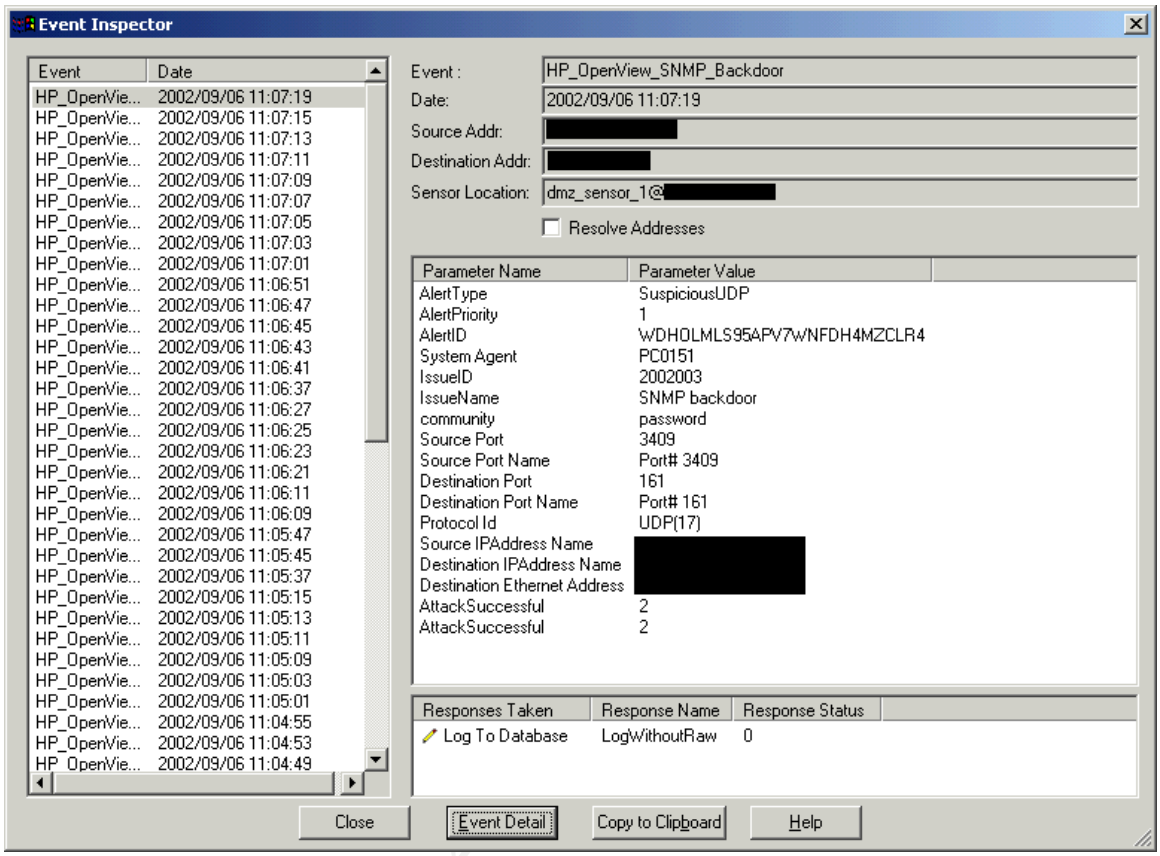

Figure 3.

The AlertID and community values were different in each of the 59 occurrences. The event detail explains that this event will be triggered by a signature that detects the use of “a specific, hidden SNMP community string that has read-write access to the configuration of HP Openview 4.X and 5.X management Agents”, on HP-UX and Solaris platforms. Since this server is a Windows 2000 machine, this alert should probably be turned off within the policy loaded on the sensor. Although, if the exploit was not checked for because it is specific to HP-UX and Solaris, there would have been no indication of any SNMP exploit attempts on the server. The alert did indicate the correct source IP address, which would aid in further investigation should it have been a real attempt, not one not caused by my own vulnerability scanning. The scan did not detect SNMP running; therefore, it was not a vulnerability listed in the results of the scan of this server.

Performing the same steps taken to investigate the HP_OpenView_SNMP_Backdoor alert, I found that the other high priority event, the FTP_Root alert was correct. This alert indicates that there was an attempt

to execute the “CWD~root” command which would give the attacker root permissions to read, write, and transfer files. The L1_Inventory scan did in fact detect to see if this vulnerability existed.
Therefore, the “out of the box” L1_Inventory scan, does a little more than just determine the OS. Displaying the properties for the L1_Inventory scan shows that it checked for all of the exploits shown in Figure 4. The exploits are listed in the right column.

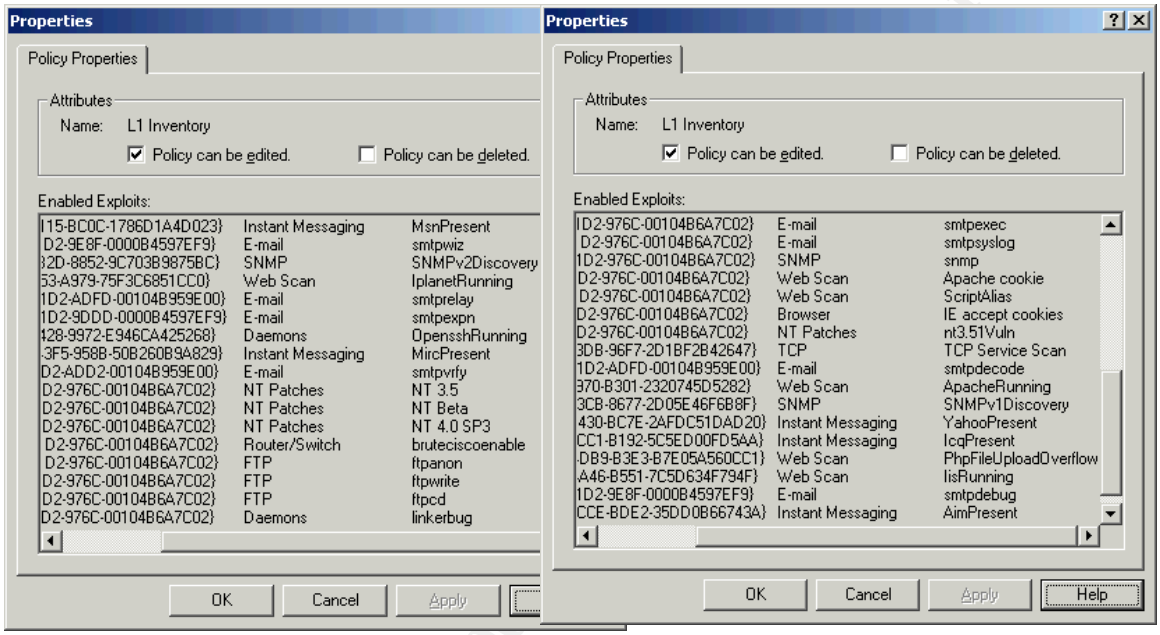

Figure 4.

After performing an L1_Inventory scan, I then ran a more specific scan of the FTP server. I started with a blank policy and modified it by turning on only FTP vulnerability checks. This was made easy by using the search feature in the policy. The search feature will seek through the 1145 exploits for anything that matches a string, which in my case was “FTP”. It came back with 166 total exploits that contained FTP somewhere in the description of the vulnerability check. This could be just the URL to Microsoft’s ftp site to download a patch; therefore, not necessarily applicable to an FTP server. Of those 166, 39 were classified as Denial of Service checks, and 139 were in the Standard category (Backdoors, E-mail, Firewall, Router, SNMP, etc.). Further investigation of each check, allowed me to remove the ones that were for an operating system other than Windows 2000, or specifically for a router, exchange server, etc. This left 21 exploits. The scan determined two low priority vulnerabilities, both of which were due to the fact that the finger service was running on the server.

One of the disadvantages of a signature based IDS is that it probably will not detect an attack that only slightly differs from the predefined set of events within the signature. To test how well Internet Scanner did in this arena, I scanned a test PC that was running a Back Orifice 2000 client. The default ports checked

for the BackdoorBo2k signature are 54340, 31337, 1025, 54321. I installed the BO2K client on port 40404. Internet Scanner failed to detect the backdoor until I configured it to specifically scan for this port. The average hacker will probably change the default ports when installing a backdoor such as this. RealSecure does provide the ability to scan all ports, but this will increase the time for the scan to complete. All scans can be configured to run via command line and therefore could be scheduled to run over night. It could even be broken down into multiple scans, each running on a different set of ports.

After

Within 24 hours of the server sensor installation on the FTP server, a high priority alert was displayed on the console, which was not due to my vulnerability scanning. The source IP of the alerts were from eight different IP addresses that are not owned by my company. The alerts indicated several attempts to exploit the HTTP_IIS_Unicode_Translation vulnerability. This signature detects HTTP_Get requests, which may indicate that a user is trying to bypass IIS security. The FTP server is not running a web server, so it is not applicable. These attempts did prompt me to modify the firewall policy to allow only ftp traffic to the FTP server in the DMZ. The firewall originally allowed both FTP and HTTP traffic. After this change was implemented on the firewall, the HTTP_IIS_Unicode_Translation alerts stopped. As of the writing of this paper, no other alerts have appeared.

The DNS server sensor has sent alerts to the console regarding Administrator Logons, event log access and additions. All were valid events triggered by local access by the server administrator. The only suspicious activity displayed so far, has been 2 low priority events: Logon_to_account_failed and Failed_login-bad_user_name_or_password. These occur every night at about the same time. The source IP is a server located in a manufacturing plant. Further investigation is necessary to resolve, but I do not believe it is of malicious intent. It is probably caused by a misconfigured application or service.

Conclusion

I believe that I have improved the integrity of my Company's overall information systems security infrastructure by implementing RealSecure and Internet Scanner. As stated in the November 1999 ITL Bulletin:

> "… an excellent approach for protecting a network may be to use an IDS to detect when an attacker has penetrated a system…Although the attacker may continue to probe the network for weaknesses, the IDS should detect theses attempts, may block these attempts, and can alert security personnel who can take appropriate action."

Another ITL Bulletin from May 1999, "Computer Attacks: What They Are and How to Defend Against Them" states "IDSs cannot be used in isolation, but must be part of a framework of computer security

measures." The bullet4in lists 14 security measures to implement to help secure a network from attack.

1. Software Patching
2. Virus Detection
3. Firewalls
4. Password Crackers
5. Encryption
6. Vulnerability Scanners
7. Configuring hosts for security
8. War dialing
9. Security Advisories
10. Intrusion Detection
11. Network Discovery Tools and Port Scanners
12. Incident Response Handling
13. Security Policies
14. Denial of Service Testing (for firewalls and web servers)

With the capabilities of RealSecure and Internet Scanner I have implemented or partially implemented seven out of the fourteen security measures (1,4,6,7,10,11,14).

After scanning the FTP server and DNS server for vulnerabilities using the ISS Internet Scanner, and then fixing those vulnerabilities, I am confident that these two servers are secure and will not allow a successful malicious attempt to disrupt any services. The next days, weeks, and months will be partially spent modifying the policy installed on the sensors to make sure I am checking for all exploits applicable to the hardware and services running on each, while at the same time not causing a significant performance degradation on the server. Continuous monitoring and updating of the sensors and scanner as new XPUs are released to combat new exploits is a must. I also will perform additional scans of other important servers, to ensure they have no vulnerabilities.

In the near future, I will hopefully be able to install network sensors in addition to more server sensors. I also have to implement an Incident Response policy to ensure proper handling of security events when they occur. "Information security requires a whole-hearted organizational commitment of resources (financial, human, and technological) to an enterprise-wide program designed to evolve and adapt to new dangers." (Power, p.3) My company is only in the beginning stages of understanding this statement. With the results and statistics I can compile from the RealSecure sensors and the firewall logs, I hope to persuade management into committing more resources to Information Security to further apply a Defense in Depth security strategy.

References:

"Acquiring and Deploying Intrusion Detection Systems." ITL Bulletin November 1999. http://www.itl.nist.gov/lab/bulletns/nov99.htm (April 21, 2002)

"Cyber crime bleeds U.S. corporations, survey shows; financial losses from attacks climb for third year in a row." Computer Security Issues press release, April 2002
http://www.gocsi.com/press/20020407.html (April 13, 2002)

Power, Richard. "2002 CSI/FBI Computer Crime and Security Survey" Computer Security Issues & Trends, Vol. VIII, No.1, Spring 2002

"Computer Attacks: What They Are and How to Defend Against Them"
ITL Bulletin May 1999
http://www.itl.nist.gov/lab/bulletns/may99.htm (April 21, 2002)

Bandy, Phil. Money, Michael. Worstell, Karen.
"Should communication between the sensor (or agent) and the monitor be encrypted?" http://www.sans.org/newlook/resources/IDFAQ/communication.htm (April 20, 2002)

Laing, Brian W. "How do you implement IDS (network based) in a heavily switched environment?"
http://www.sans.org/newlook/resources/IDFAQ/switched.htm (April 20, 2002)

" About ISS." Internet Security Systems
http://www.iss.net/about/ (August 18, 2002)

Real Secure 6.5 Student Guide. Atlanta, GA:
Internet Security Systems, 3/26/02

Internet Scanner 6.2 Student Guide, Atlanta, GA:
Internet Security Systems, 5/9/02