Seth Law
GSEC Practical Assignment
Version 1.4b

Low Cost Network Security

Introduction

Even though many small and medium businesses lack the funds to install proprietary network tools for intrusion detection, vulnerability scanning, and network analysis, there are options available from the open source community that can fill these security needs. Since security in-depth is a concern that any business with an Internet presence should practice, these tools need to be explored and implemented. With a little ingenuity and time, it is possible to create a high-class security infrastructure at little cost. This paper concentrates on integrating open source tools and leveraging them into a security infrastructure that is easy to use and maintain.

Tools

There is a wide selection of open source tools available that are useful in deploying a security infrastructure. This paper explores the following: Snort and ACID for use in network intrusion detection, Nmap and Nessus for use in vulnerability scanning and detection, and Ntop for use in anomaly detection.

Prerequisites

Some of these tools (Nessus) are only available for unix type operating systems. Therefore, this paper will assume that all tools are being implemented on one system that acts as a central security system. It would be trivial to replace the operating system with a unix flavor of your choice, but make sure the tools are supported on that operating system before installing.

The central security system requires an Apache web server (http://httpd.apache.org) with PHP (http://www.php.net) installed as well as some sort of database (MySQL http://www.mysql.com or PostgreSQL http://www.postgresql.org are good open source options). The web server and database are needed by Snort and ACID and will be mentioned accordingly. The web server is an important piece that provides a central location for a consolidated view of the security status of the network.

Because of the nature of analysis and logging, these tools require processing power and disk space. When monitoring a high-traffic system, powerful hardware is required for the system to perform effectively.

Snort (http://www.snort.org)

Snort is described as “an open source network intrusion detection system, capable of performing real-time traffic analysis and packet logging on IP networks” (Caswell). Snort is a system that watches all traffic on the network for known “bad” patterns and records these attempts in a chosen manner.

Snort also “allows for raw packet data analysis. This allows for examination of a packet down to the payload to determine what caused the alert, why […] something caused the alert, and whether action needs to be taken” (Brennan). This low level view of an attack increases a security administrator's knowledge of what constitutes an attack and how to better defend against them.

Being able to recognize known bad traffic on a network is necessary to secure a network. Snort's database of attack signatures is continually updated by the open source community. The syntax of these rules is simple, allowing a security administrator to customize rules for the protected network. With the constant flow of attacks, it is necessary for the security administrator to update the local instance of Snort to use the latest attack signatures.

The Snort website (http://www.snort.org/docs/) has some good tutorials for installing Snort on Linux, FreeBSD, or Windows 2000 as well as white papers and the official Snort documentation.

For the purposes of this paper, Snort must be configured to log all alerts and information to the database (see the Snort user manual on how this is accomplished). ACID will need this later to analyze the alert data.

ACID (http://www.andrew.cmu.edu/~rdanyliw/snort/snortacid.html)

ACID is an acronym for Analysis Console for Intrusion Databases. It is described as “a PHP-based analysis engine to search and process a database of security events generated by various IDSes, firewalls, and network monitoring tools” (Danyliw). This is the system that will show the current attack trends on your network, both graphically and with text.

It is important to have some system in place to help a security administrator identify both current and historical attacks. This historical data gives an excellent view of where and when attacks occurred. Without this view, it would be difficult to distinguish between the alerts from real attacks and false positives.

Nmap (http://www.insecure.org/nmap)

Nmap is a shortened name from Network Mapper. It is “an open source utility for network exploration or security auditing” that “uses raw IP packets in novel ways to determine what hosts are available on the network, what services (ports) they are offering, what operating system (and OS version) they are running, what type of packet filters/firewalls are in use, and dozens of other characteristics" (Fyodor). This is a highly useful utility that both security administrators and crackers use to determine the layout of a network.

A good explanation of Nmap is that it "will scan a range of host addresses or a network address range entered at the command line. It will determine which addresses are active systems currently on line. It will probe a range of ports, selectable by the user, to see what services the identified system is running. Finally it will probe the system for responses to some unusual packets to try and guess what operating system is installed on the target system" (Corcoran).

Nessus (http://www.nessus.org)

Nessus is "security scanner […] software which will audit remotely a given network and determine whether bad guys (aka 'crackers') may break into it, or misuse it in some way" (Deraison). First, Nessus takes the output from Nmap and cross references the services with known vulnerabilities. Then it tests each service to see if they are exploitable and reports on the results. Nessus does the same work that any intruder would to exploit the system. When "first scanning yourself using [an intruder's] tool of choice, you are taking the initiative in preventing the exploitation, since you are aware of what they are looking for and have already taken steps to prevent it" (Christensen).

Nessus is also useful because it presents the vulnerability data in a human readable form. When performing a scan, use the "-T html" option to display the data in html form in the specified file. This file will be used in analyzing and preventing intrusions.

Because a network is constantly changing, Nessus should be setup to scan the entire network at regular intervals. This gives the security administrator knowledge about any new services that may crop up, as well as historical data that can be used in analyzing new vulnerabilities. Even if it is not possible to patch the vulnerability as soon as it is evident, knowledge of its existence means it can be monitored.

To secure a network in-depth, a security administrator must understand the computers and services that are legitimate and know when things change. By keeping historical vulnerability scanning data, it is a simple matter to compare current network status to a known good network status.

Ntop (http://www.ntop.org/ntop.html)

Ntop "is a network traffic probe that shows the network usage" (Deri). This utility provides a view of all the traffic on your network, as opposed to Snort that only monitors for traffic that it finds interesting. It records the number of packets and the amount of data sent and compiles this information into an overall view of network usage. This view shows the security administrator the strange traffic on the network and provides an opportunity to track down anomalous behavior.

Ntop also provides a useful human readable web interface that can be used to track down anomalies and previously unknown attacks.

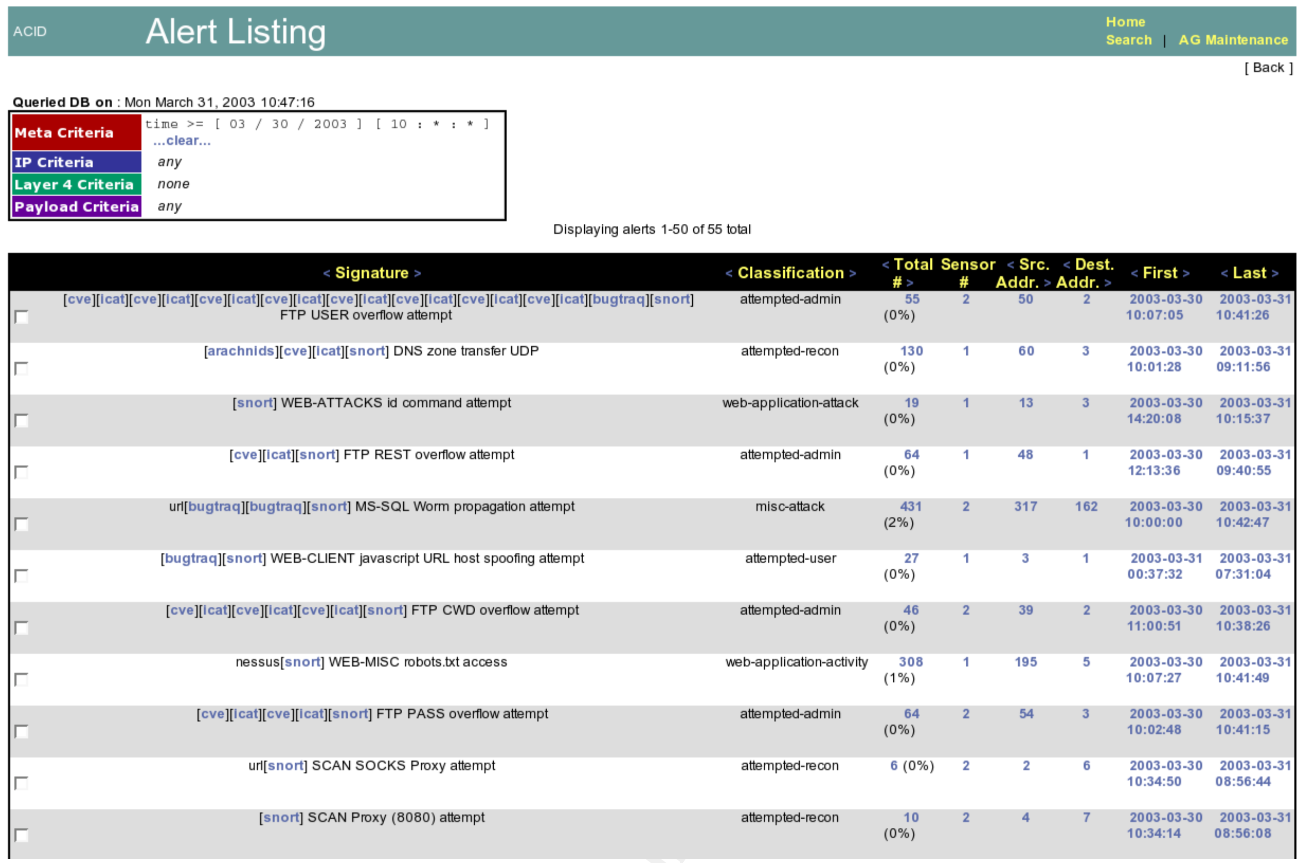
Analysis

It is important to understand what data each tool provides and how to analyze the data from a security perspective. In this section network intrusion detection, anomaly detection, and vulnerability scanning will each be discussed in turn. A security administrator must understand the data coming from each of the aforementioned tools and determine if an attack is or has taken place and whether notification of security administrators is necessary.

Intrusion Detection

ACID is the front end interface that makes sense of the alerts and warnings that Snort provides. Without understanding the data Snort provides, it would be impossible to analyze the intrusion detection data. ACID has powerful data mining capabilities that can be leveraged. Read the document on "Searching and Specifying Criteria" located on the ACID website at http://www.andrew.cmu.edu/~rdanyliw/snort/acid_instruct.html for a complete tutorial on how to use its searching and reporting capabilities.

A security administrator should check the status of alerts daily and determine what attacks (if any) have occurred and determine the action warranted by those attacks. ACID provides reports that show the alerts generated over the last 24 hours or the last week. The first report is found on the front page of ACID under the *Snapshot* heading. Follow the *unique* link after *Last 24 hours*. This table shows all unique alerts logged by Snort in the last 24 hours.



ACID **Alert Listing**

Home
Search | AG Maintenance

[Back]

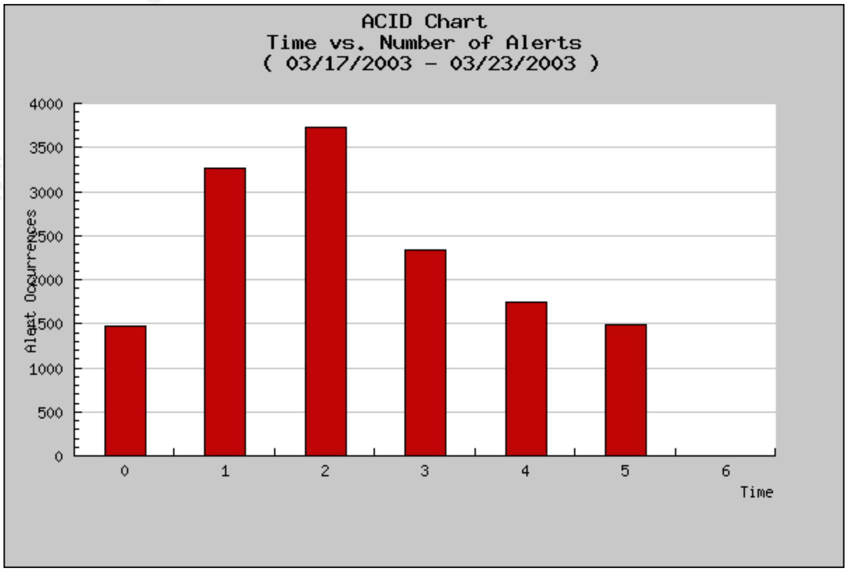
Queried DB on : Mon March 31, 2003 10:47:16

Meta Criteria	time >= [03 / 30 / 2003] [10 : * : *] ...clear...
IP Criteria	*any*
Layer 4 Criteria	*none*
Payload Criteria	*any*

Displaying alerts 1-50 of 55 total

	< Signature >	< Classification >	< Total # >	Sensor #	< Src. Addr. >	< Dest. Addr. >	< First >	< Last >
□	[cve][icat][cve][icat][cve][icat][cve][icat][cve][icat][cve][icat][cve][icat][cve][icat][bugtraq][snort] FTP USER overflow attempt	attempted-admin	55 (0%)	2	50	2	2003-03-30 10:07:05	2003-03-31 10:41:26
□	[arachnids][cve][icat][snort] DNS zone transfer UDP	attempted-recon	130 (0%)	1	60	3	2003-03-30 10:01:28	2003-03-31 09:11:56
□	[snort] WEB-ATTACKS id command attempt	web-application-attack	19 (0%)	1	13	3	2003-03-30 14:20:08	2003-03-31 10:15:37
□	[cve][icat][snort] FTP REST overflow attempt	attempted-admin	64 (0%)	1	48	1	2003-03-30 12:13:36	2003-03-31 09:40:55
□	url[bugtraq][bugtraq][snort] MS-SQL Worm propagation attempt	misc-attack	431 (2%)	2	317	162	2003-03-30 10:00:00	2003-03-31 10:42:47
□	[bugtraq][snort] WEB-CLIENT javascript URL host spoofing attempt	attempted-user	27 (0%)	1	3	1	2003-03-31 00:37:32	2003-03-31 07:31:04
□	[cve][icat][cve][icat][cve][icat][snort] FTP CWD overflow attempt	attempted-admin	46 (0%)	2	39	2	2003-03-30 11:00:51	2003-03-31 10:38:26
□	nessus[snort] WEB-MISC robots.txt access	web-application-activity	308 (1%)	1	195	5	2003-03-30 10:07:27	2003-03-31 10:41:49
□	[cve][icat][cve][icat][snort] FTP PASS overflow attempt	attempted-admin	64 (0%)	2	54	3	2003-03-30 10:02:48	2003-03-31 10:41:15
□	url[snort] SCAN SOCKS Proxy attempt	attempted-recon	6 (0%)	2	2	6	2003-03-30 10:34:50	2003-03-31 08:56:44
□	[snort] SCAN Proxy (8080) attempt	attempted-recon	10 (0%)	2	4	7	2003-03-30 10:34:14	2003-03-31 08:56:08

The second report to be checked daily is a weekly report of the number of alerts. To access this information, use the "Graph Alert Data" utility listed on the ACID main page. Map the number of alerts per day for the last week. To do this, select a Chart Type of "Time (day) vs. Number of Alerts" with the "Chart Begin" set to 7 days previous and "Chart End" set to the current day. The bar graph seems to be an easy way to interpret if the number of attacks increased over the last week. The following graph is an example of one such report.



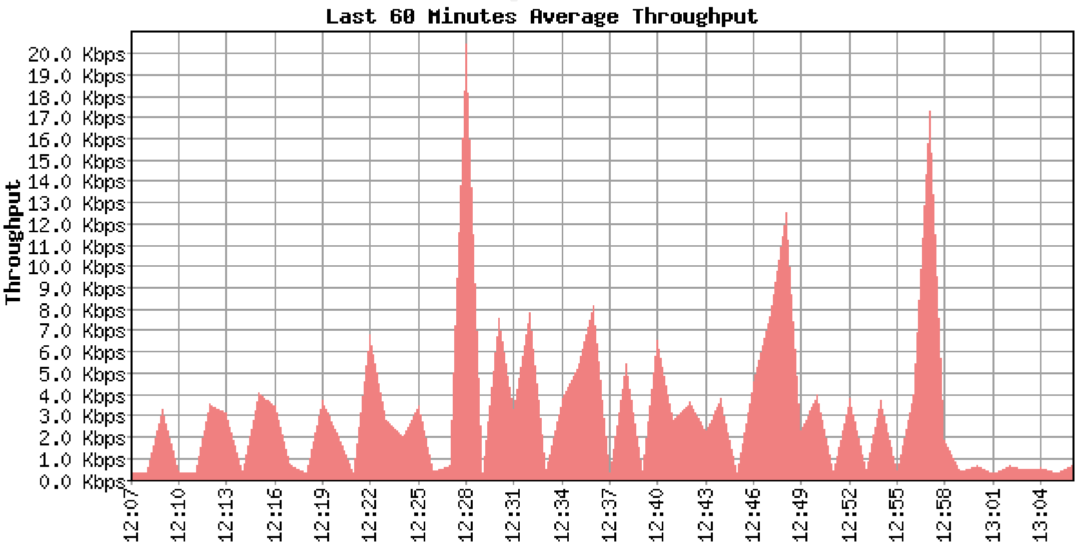
Any spikes or anomalies in this number should be investigated to determine whether an intrusion attempt occurred. When the number of attacks on any day is much greater than other days, check the unique alert listings to see what attacks were the most common. A great number of attacks from one IP address usually results from an automated attack or probe.

Anomaly Detection

Ntop provides a summary of all network traffic being generated. This “network knowledge simplifies the task of specifying the rules for defining burglar alarms” (Deri and Suin). Using Ntop makes the task of Snort and the IDS easier by allowing Ntop to watch all traffic that does not match a pattern in Snort.

One good example of a use for Ntop was the recent SQL Slammer virus. Since the virus spread very quickly, it was impossible to have a previous rule in an IDS to detect its traffic. Once a network or security administrator noticed that the network load had spiked, it would have been a simple operation to use Ntop to track down the type of traffic that was causing the load.

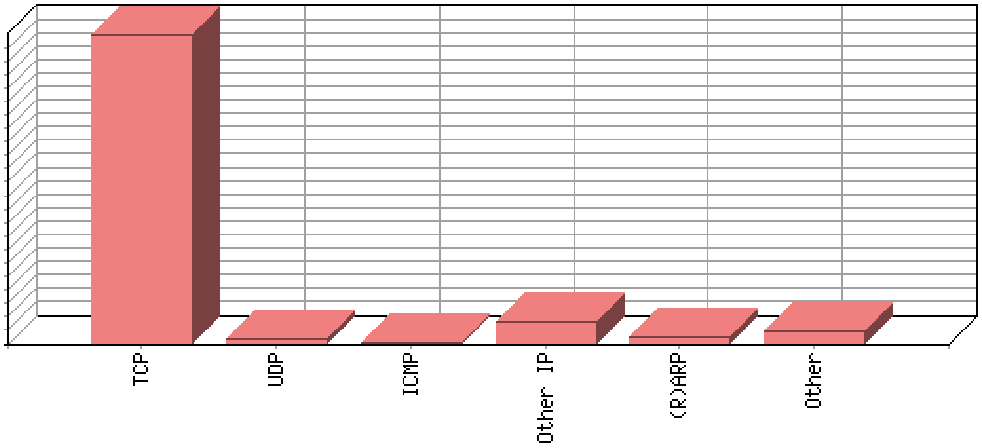
The main page of Ntop (located on port 3000 by default) gives a brief explanation of what it is doing, but useful information is located under the Stats tab. After clicking on Stats, follow the link on the left for Network Load. This shows the network load over the last hour and the last 24 hours graphically. The following image shows traffic on a small network over the last hour.



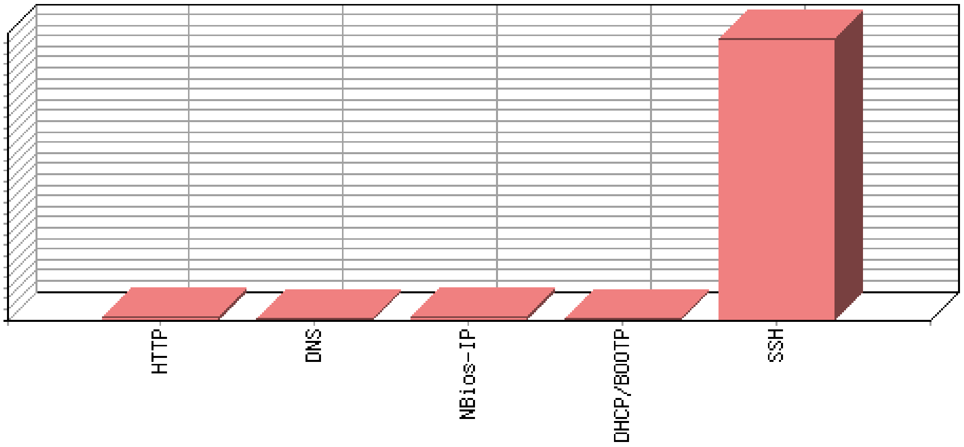
This graph shows low activity on the network. A couple of spikes show possible web browsing or file downloads. In a case such as SQL Slammer, the graph would show increased activity at a much higher rate. Click on the graph to do a deeper analysis of the traffic that has been flowing across the network in this timeframe.

The other useful link under the Stats page is the Traffic link. Follow this link to get a detailed analysis of all traffic on the network since Ntop has been sampling. Two of the most useful generated graphs are located at the bottom of the page under the titles *Global Protocol Distribution* and *Global TCP/UDP Protocol Distribution*.

Global Protocol Distribution



Global TCP/UDP Protocol Distribution

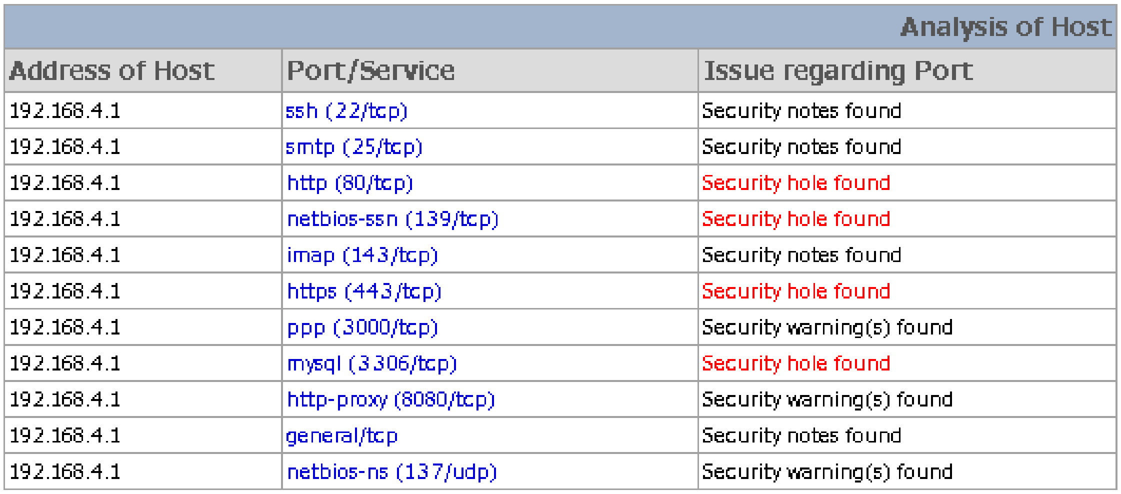


The *Global Protocol Distribution* graph shows that most of the traffic was TCP based. The *Global TCP/UDP Protocol Distribution* gives a more detailed analysis of the TCP and UDP traffic. In this case, a large portion of the TCP traffic is over the ssh port (22), indicating that a secure shell session is traveling across the network that is being monitored. If this service is not allowed according to the network policy, any traffic of this type would be cause for concern. To refer back to an earlier example, if SQL Slammer were currently active on this network, the graphs would show more UDP activity on the MS-SQL port.

To dig even further into the hosts that are generating all of the traffic, use the Hosts link on the left. This will help to determine which hosts are generating interesting traffic.

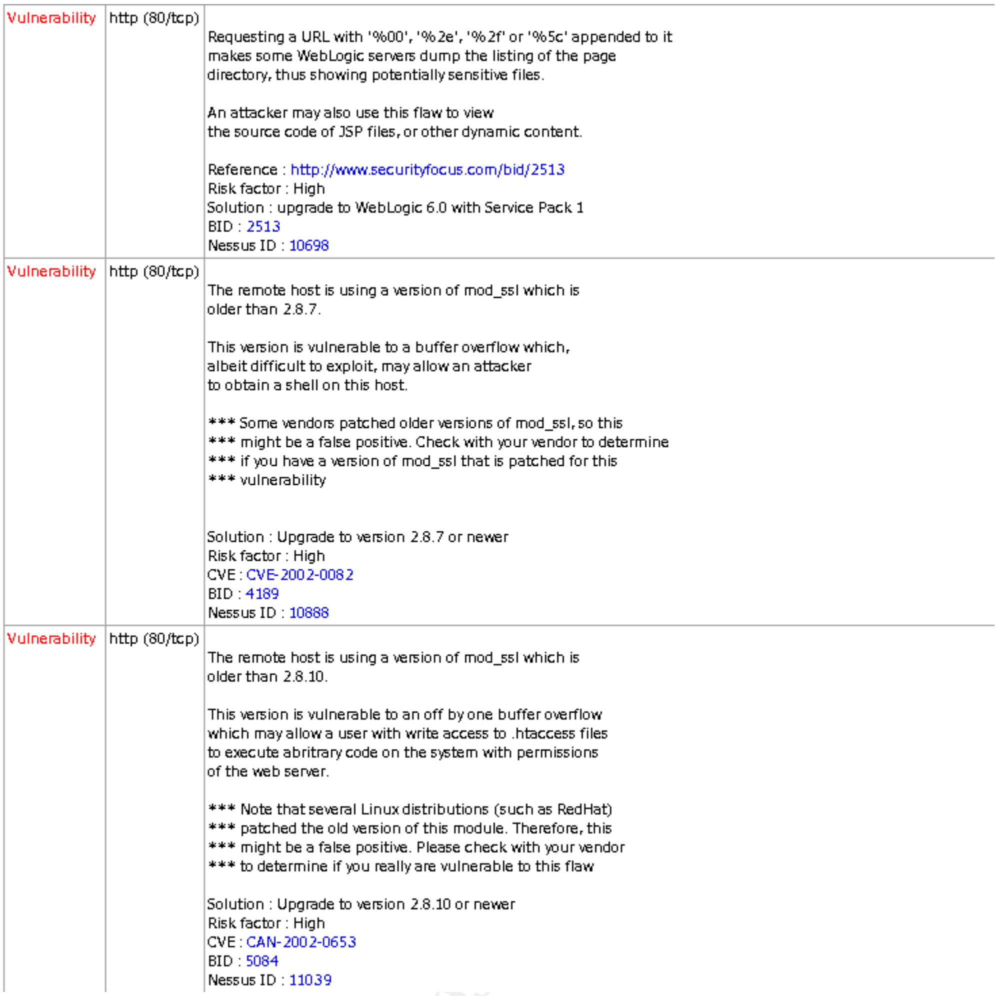
Vulnerability Scanning

Now that attacks and anomalies are detected on the network, it is important to try and minimize the vulnerabilities that exist on the network. Nessus and Nmap provide this function. This portion of network security can involve the most time from day to day, since it involves tracking down and patching vulnerabilities. The -T html option of Nessus (run from the command line) provides an easily viewed web page that lists the open ports of an IP address (the Nmap output), and any known exploits that are associated with that port. Here is a sample of Nessus output:



		Analysis of Host
Address of Host	**Port/Service**	**Issue regarding Port**
192.168.4.1	ssh (22/tcp)	Security notes found
192.168.4.1	smtp (25/tcp)	Security notes found
192.168.4.1	http (80/tcp)	Security hole found
192.168.4.1	netbios-ssn (139/tcp)	Security hole found
192.168.4.1	imap (143/tcp)	Security notes found
192.168.4.1	https (443/tcp)	Security hole found
192.168.4.1	ppp (3000/tcp)	Security warning(s) found
192.168.4.1	mysql (3306/tcp)	Security hole found
192.168.4.1	http-proxy (8080/tcp)	Security warning(s) found
192.168.4.1	general/tcp	Security notes found
192.168.4.1	netbios-ns (137/udp)	Security warning(s) found

This list is similar to the output given by Nmap, but adds the fact that certain ports have security holes associated with them. In reviewing the security holes associated with our web server (port 80), we get the following output:



Vulnerability	http (80/tcp)	Requesting a URL with '%00', '%2e', '%2f' or '%5c' appended to it makes some WebLogic servers dump the listing of the page directory, thus showing potentially sensitive files. An attacker may also use this flaw to view the source code of JSP files, or other dynamic content. Reference : http://www.securityfocus.com/bid/2513 Risk factor : High Solution : upgrade to WebLogic 6.0 with Service Pack 1 BID : 2513 Nessus ID : 10698
Vulnerability	http (80/tcp)	The remote host is using a version of mod_ssl which is older than 2.8.7. This version is vulnerable to a buffer overflow which, albeit difficult to exploit, may allow an attacker to obtain a shell on this host. *** Some vendors patched older versions of mod_ssl, so this *** might be a false positive. Check with your vendor to determine *** if you have a version of mod_ssl that is patched for this *** vulnerability Solution : Upgrade to version 2.8.7 or newer Risk factor : High CVE : CVE-2002-0082 BID : 4189 Nessus ID : 10888
Vulnerability	http (80/tcp)	The remote host is using a version of mod_ssl which is older than 2.8.10. This version is vulnerable to an off by one buffer overflow which may allow a user with write access to .htaccess files to execute abritrary code on the system with permissions of the web server. *** Note that several Linux distributions (such as RedHat) *** patched the old version of this module. Therefore, this *** might be a false positive. Please check with your vendor *** to determine if you really are vulnerable to this flaw Solution : Upgrade to version 2.8.10 or newer Risk factor : High CVE : CAN-2002-0653 BID : 5084 Nessus ID : 11039

Armed with a list of possible vulnerabilities, each will be addressed in turn. The first vulnerability can be dismissed since the web server currently running is Apache, not Weblogic. The second and third are cause for greater concern. In doing some research, it is determined that the server is running an old version of mod_ssl, so an upgrade for this machine needs to be scheduled as part of maintenance.

Researching each security hole that Nessus reports can be a tedious and time consuming process, but it is a worthwhile activity. First, it gives a network administrator a feel for all the systems that are running on the network and what services they provide. Second, it educates a security administrator about the holes that exist in different applications and which Snort signatures are useful for the network being monitored. Finally, as long as Nessus is kept up to date, it keeps the security administrator informed as to the latest attacks and exploits, so he can identify attacks coming from outside the network.

Notification

Now that all network traffic is being monitored, some system is needed to notify the security administrators when the network is being attacked.

One popular utility called Pigsentry (http://www.proetus.com/tools/pigsentry) is a perl script that maintains a state table of Snort's alerts and notifies an administrator when new alerts are generated. To implement this or any other type of notification system, it is imperative to optimize the signature set that Snort is using so that false positives don't fill up an email box or pager and all notifications get ignored.

Other notification systems exist for network availability. Nagios (http://www.nagios.org) is a popular open source system that also monitors system availability. Nagios is out of the scope of this paper and will not be discussed in detail.

Integration

All of these tools focus on one specific need as far as security is concerned. When integrated together, they display an excellent view of interesting security events on a network. With an understanding of what these tools provide, it is simple to create a web page that will direct an administrator to the relevant information from each tool. This page can be as complex or as simple as the administrator wishes to make it.

One important issue with a central web repository is protection. The information would provide an intruder a basic understanding of a network better than most employees. At the very least, this information should only be accessible on a need-to-know basis. Maintain a user database of authorized users and apply the user authentication methods provided by Apache. It would also be worthwhile to encrypt the data using SSL. The more steps taken to prevent discovery of this repository by an intruder adds up to more time for a security administrator to uncover the intrusion and avoid further casualties.

A simple web page with a minimum amount of links would contain a link to the main page of both ACID and Ntop as well as the latest Nessus scan. It is useful to follow the instructions at http://www.ntop.org/UsageNotes.html on how to proxy Ntop through Apache. This makes it easier to provide one interface to all information and to protect the Ntop application from being exploited.

To create a more useful interface, include both the *Network Load Statistics* and *Global Traffic Statistics* that are displayed by Ntop and a link into the directory that contains all Nessus scans. In this case it is necessary to script some sort of file rotation for Nessus so that the last scanned file will not be overwritten when a scan is performed. It is also possible to construct this page using PHP or Perl to include a link into ACID for the last 24 hours of alerts and graphing out the last 7 days of alerts. While any of these links are easy to navigate to using each application, by presenting them only one click away it is easier for a security administrator to get a quick idea and save time each day in evaluating the current state of network security.

Maintenance

Now that the security system is setup and being checked daily, it is important that the system remain up-to-date. It is not possible to discover intrusions that exploit the latest vulnerabilities if the detection system does not know about them.

One of the most important aspects of maintenance is time. Spending 30 minutes a week to get the latest Snort signatures and Nessus probes installed can cut the time needed to clean up after an intrusion. It may even prevent the intrusion from ever occurring.

Check for new Snort signatures at least once a week. These can be found at http://www.snort.org/dl/rules. Make sure to customize the rules to reflect the current network and replace the last set of rules completely. Snort itself is also being updated and optimized. Watch for new versions and take the time to implement them. A new version often contains security updates and enhancements that will improve network intrusion detection.

New Nessus plugins are easily installed by using the *nessus-update-plugins* program that installs as part of Nessus. This script will download the latest plugins for Nessus and implement them into the scanning infrastructure. The latest plugins are also available manually at http://www.nessus.org/scripts.php.

ACID, Nmap, and Ntop are all being actively developed and should be upgraded when new versions are available, but a daily or weekly check of the websites is not imperative as long as a security administrator is subscribed to Bugtraq (http://www.securityfocus.com/popups/forums/bugtraq/intro.shtml) or another security mailing list. These lists provide knowledge of new releases and any security vulnerabilities of these products.

This security system is also expandable. Where it can be implemented on a single server, it may also be divided into multiple servers for each function. One server providing intrusion and anomaly detection using Snort and Ntop, a second providing vulnerability scanning using Nessus and Nmap, and a last server hosting a central database and web server for reporting. Depending on the size and load of the network multiple intrusion and anomaly detection sensors may be needed.

Conclusion

.

Network security is a crucial aspect of business in today's world. Intrusion detection, vulnerability scanning, and anomaly detection each provide a different layer of the overall network security. Even if a security administrator does not have the budget for the expensive proprietary network security solutions, security in-depth can still be gained by applying the preceding ideas and open source tools to build an integrated security solution.

References

Brennan, Michael P. “Using Snort For a Distributed Intrusion Detection System” 29 January 2002. URL: http://www.sans.org/rr/intrusion/distributed_IDS.php (19 Apr. 2003).

Caswell, Brian. "Snort - The Open Source Network IDS: More info about Snort" 21 April 2003. URL: http://www.snort.org/about.html (21 Apr. 2003).

Christensen, Paul. “An Introduction to Nessus” 7 May 2001. URL: http://www.linuxsecurity.com/feature_stories/feature_story-86.html (19 Apr. 2003).

Corcoran, Tim. “An Introduction to NMAP” 25 October 2001. URL: http://www.sans.org/rr/audit/nmap2.php (17 Apr. 2003).

Danyliw, Roman. “Analysis Console for Intrusion Databases (ACID)” URL: http://www.andrew.cmu.edu/~rdanyliw/snort/snortacid.html (18 Apr. 2003).

Deraison, Renaud. “The Nessus Project: Introduction” URL: http://www.nessus.org/intro.html (20 Apr. 2003).

Deri, Luci. “ntop – network top” URL: http://www.ntop.org/overview.html(21 Apr. 2003).

Deri, Luci and Suin, Stefano. “Improving Network Security Using Ntop” URL: http://jake.unipi.it/~deri/RAID.pdf (20 Apr. 2003).

Fyodor. “Nmap – Free Stealth Port Scanner For Network Exploration & Security Audits. Runs on Linux/Windows/UNIX/Solaris/FreeBSD/OpenBSD” 19 April 2003. URL: http://www.insecure.org/nmap (20 Apr. 2003).