Global Information Assurance Certification Paper

SANS/GSEC Practical Assignment
Security Essentials
Secure Setup of a Corporate Detection and Scanning Environment

GSEC Assignment: 1.4b
Submitted by: Dieter Sarrazyn
Submitted on: 22/08/2003

Table of Contents

List of Figures

List of Tables

1. Introduction

This paper covers the secure deployment of a distributed intrusion detection environment as well as the secure deployment of a distributed vulnerability scanning environment.

Since a lot of companies do not have the proper security budget (yet), the focus of this paper lays on using open source tools. The open source tools that are used are snort (with acid), nessus, nmap, nikto, inprotect and gherkin. The URL's of the websites of these tools can be found in the reference list and in the components description.

The different chapters in this document are:

- Components description

  This chapter contains some brief descriptions of the products and tools that are used in this paper.

- Network Setup

  The network setup describes how the different components can be placed securely in a corporate network. This is done with centralized management, log consolidation and secure communications between the systems. This chapter also covers network related security measures that need to be in place on the different components of the network.

- Components Setup

  This chapter describes the setup of the different components used in the secure setup.

  The central management server will be responsible for the management of all the components. Second, this system will be used as a log consolidation machine (using syslog & the log functionality of the tools used) and as a central time server for all sensors and scanners deployed.

  Another part in this chapter is the installation, configuration and deployment of a Distributed IDS Environment (based on snort and acid) and a Distributed scanner system (based on nessus and nmap) in a secure way.

- Analysis of results

  Analysis of the results obtained through the different IDS sensors, the several vulnerability scanners and port scanners have to be analyzed to be useful for the company. This is also the case for the centralized syslog events.

- Conclusions

  This chapter contains the final conclusions of this paper.

2. Conventions

Following are the typographic conventions I used for this paper.

Regular text is in “Arial”, 12 points (as defined by the Assignment v1.4b and Administrativa v2.5b)

Source code, program-output and commands are in `“Courier New”, 9 points and shaded` to distinguish between code/output/commands and regular text.

3. Components description

3.1. Snort

Snort is a open source Network Intrusion Detection System (NIDS) and is available from http://www.snort.org

Snort uses a central configuration file (called snort.conf) where all variables can be configured needed for the correct working of snort. It uses various rules files which contain all signatures for all known network based attacks. These rules and signatures are updated regularly as knew vulnerabilities are found daily. The data logging of snort can be done using a database (mysql, postgres…), using tcpdump files, using the syslog facilities and by using various other methods.

3.2. ACID

ACID (Analysis Console for Intrusion Databases) is a web based log analysis tool which can be used to analyse the events that snort logged to the database. Acid is available from the following website:
http://www.andrew.cmu.edu/~rdanyliw/snort/snortacid.html

Extensive filtering is possible through the web interface so that log analysis becomes an easy task.

3.3. SnortCenter

SnortCenter is a central management solution for distributed snort sensors. It can be found on http://users.pandora.be/larc/

SnortCenter also uses a database (mysql) to store all configuration files, rules and classifications related to the snort intrusion detection system. This application consists of a server component and of agents running on the different snort sensor systems. The communication between both components can be secured by means of ssl.

3.4. Nessus

Nessus (http://www.nessus.org) is a free (open source) vulnerability scanning tool. Nessus uses several plugins to launch its tests and security checks. These plugins are also regularly updated as security vulnerabilities are found daily.

3.5. Nmap & rnmap

Nmap (http://www.insecure.org) is probably the best port scanning tool currently available. Various types of portscanning can be performed (tcp port scan, syn scan, udp port scan…) as well as operating system scanning and protocol scanning.

Rnmap is a client/server application which allows the network administrator to run port scans on a remote system. This application can be used to build up a centralized port scanning environment. Rnmap can be found on http://rnmap.sourceforge.net/.

3.6. Scanner Web interfaces

Inprotect (http://www.inprotect.com) is a web interface for nessus and nmap. This allows you to launch nessus- and nmap-scans by using only a web browser. The data of this web interface is stored in a mysql database.

Gherkin (http://www.altmode.com/gherkin/) is another web interface solution for nessus and nmap. This solution uses a postgres database but will support mysql in the future.

Nmapwebfe is a dedicated web interface for nmap. This web interface can handle almost all of the options and settings of nmap. Nmapwebfe can be found on http://davidquintana.com/projects/ No information is stored in a database here.

3.7. Mod_security plugin for apache

According to Ivan Ristic, the creator of Mod_Security: “Mod_Security is an open source intrusion detection and prevention engine for web applications. It operates embedded into the web server, acting as a powerful umbrella - shielding applications from attacks.”

According to the manual written by Ivan Ristic, some of the features of mod_security are; Request filtering; Anti-evasion techniques; Understanding of the http protocol; POST payload analysis; Audit Logging; HTTPS filtering

This apache security module will be used to protect all of the web enabled applications, used to remotely manage all of the different components.

The mod_security plugin can be found on http://www.modsecurity.org/.

3.8. Syslog-NG

Syslog-ng is a replacement for the regular syslog daemon. With this replacement, more security and functionality is possible as well as centralized logging towards a

mysql database. Syslog-ng can be found on http://www.balabit.com/products/syslog_ng/.

3.9. Iptables (netfilter)

Iptables is a free (open source) statefull inspection firewall solution with extensive filtering possibilities. Iptables can be found on http://www.netfilter.org.

4. Network Setup

4.1. General network setup

Below, you can find the network setup of how ids sensors and vulnerability scanners can be deployed in a secure way into your network(s).

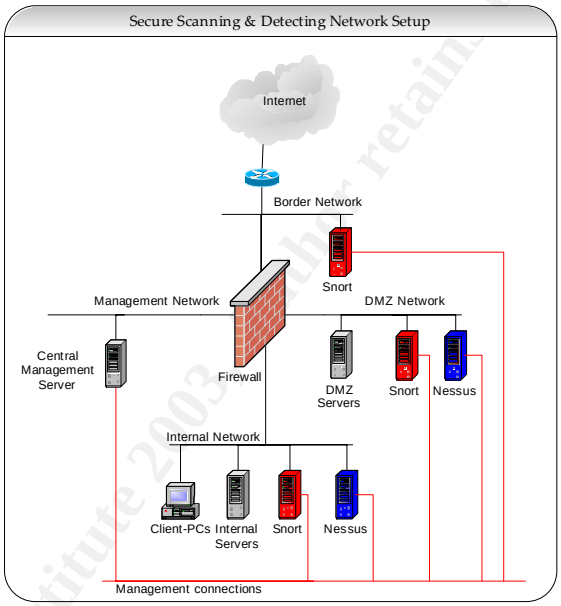

Figure 1: Network Setup

The important thing in this picture is the separate network used to communicate between the central management server and the different ids sensors and vulnerability scanners. This network is drawn in the color red.

This separated network is dedicated only for the following communication paths:

- Remote management connections (using ssh)
- Time synchronization (using ntp)
- Central logging (using syslog)
- Remote control connections for the ids sensors
- Remote control connections for the scanners

This dedicated network should ideally be placed on a separate switch. If this is not possible, a highly secured and controlled vlan could also be an option. This

network should definitely NOT be connected to the internal network since network bridges could be created and the firewall could be bypassed.

In the picture is also shown that all sensor and all scanner machines have two network interfaces each. One interface is used for the previously mentioned management communication network; the other interface is used to perform "the real work". For the IDS sensors, this interface will not have an ip address assigned (will be configured in promiscuous mode – using "PROMISC=yes" in the ifcfg-ethX configuration file - or will have a bogus ip address like 0.0.0.1/0.0.0.0) since the network sensor will put the interface into promiscuous mode. For the scanners, this interface will have a valid ip address assigned (valid for the network the interfaces resides in).
Physical network connection requirements for the network sensors are a connection onto a span port (if the physical connection happens to a switch) or onto a hub. Physical network connection requirement for the network scanner is a regular switch or hub port.

Detection if other hosts are connecting (physically) to the management connections network can be performed by running arpwatch in the central management server. Arpwatch detects all new mac addresses in the network and alerts the network administrator in case a new station is added or in case a station has changed mac addresses. Arpwatch is included in the RedHat Linux distribution or can be downloaded from http://www-nrg.ee.lbl.gov/. Alerting of the network administration can be done by means of email or through the syslog facility.

In the picture, one can also see that there is a dedicated management DMZ as well. The two dedicated management networks are used for the following reasons:

- To prevent the creation of "bridges" from the internal network towards the other networks.
- To prevent abuse of the management infrastructure by internal users
- To strictly control access towards the management server.

Not shown in the picture is a snort sensor on the dedicated management DMZ network. This snort sensor is needed to detect attacks against the management server. This additional snort sensor can be run on the management server itself. This reduces the need for an additional station.

4.2. Firewall Filtering

In order to increase the security of the intrusion detection and vulnerability scanning solution shown in the picture in the previous paragraph, the central firewall should be properly configured with the appropriate rules.

All communication between the internet and the different sensors – for example when the snort signatures are updated or when the nessus plugins are updated – should pass via the central management server. These sensors and scanners should not be allowed to connect to the internet directly since this could increase

the risk of getting these systems compromised. The central management server is the only system that is allowed to go to the internet to perform the following:

- Download new snort rules,
- Download new nessus plugins,
- Download updates for the tools used,
- Download OS updates,
- Perform time synchronization,
- Performing dns lookups

If one is really paranoid about security, one could choose to not let the management server connect to the internet but to upload the different updates from the internal network (network administrator machine), using a secure way of uploading. An option here would be using secure copy which is included into the open source implementation of Ssh – openssh – and into various free Ssh clients.

The only connection that is allowed towards the management server (not including the separated management communication network) is from the network administrator's machine, using secure shell (Ssh) and https to do so. If the network administrator has to connect to the management server with a database client, the traffic should be tunneled through secure shell using port forwarding.

An additional layer of security on the firewall system would be the requirement for authentication if the network administrator wants to connect to the central management station. Depending on the firewall solution that is used, various authentication methods can be used. The preferred authentication method is using strong authentication of course.

To summarize the above mentioned ideas and guidelines, we can create the following additional rulebase that the firewall should have installed, next to the secure rulebase that was already in place.

Source	Destination	Service	Action
Snort Sensors	Internet	Any	Drop
Nessus Scanners	Internet	Any	Drop
Management Server	snort site, nessus site	ftp, http, https	Allow
Management Server	Predefined time servers	ntp	Allow
Management Server	Predefined dns servers	dns	Allow
Network Admin machine	Management Server	ssh, https	Allow

Table 1: Main Firewall additional rulebase

4.3. Sensor Filtering & Scanner filtering

In providing additional security and especially on the management communication network, filtering is also needed on the level of the different sensor and scanner machines. The idea is that only the management server should be able to connect to the different sensor and scanner systems. Additionally, this should only be possible through the separate dedicated management communication network.

The additional level of security can be provided by means of installing a firewall solution on the sensor systems or scanner systems itself. The firewall to be used here can be netfilter/iptables (which is included in most Linux distributions) or an equivalent.

The filtering for the snort sensor systems needs to be done on the management connection interface only. One could say that there is no need for filtering on the second interface in this system since there will no ip address assigned to this interface (see further in this document) and that therefore no connections can be made to this system using this interface. However, it is also a very good idea to put an additional filter on the sniffing interface (the sensor interface) as well. This filter will be used if an attacker could compromise the snort system (using an unknown vulnerability in the snort system and entering the sensor through the sniffing interface).

The rulebase for a snort sensor, communicating with an acid central console, a central snort management and a central logging solution looks like this:

Source	Destination	Service	Action
Snort Sensor	Management Server	ntp, syslog	Allow
Snort Sensor	Management Server	mysql	Allow
Management Server	Snort Sensor	ssh	Allow
Management Server	Snort Sensor	snort management	Allow

Table 2: Sensor Firewall Rulebase

The rulebase installed on the sniffing interface looks like this:

Source	Destination	Service	Action
Snort Sensor – scanning interface	All	All	Drop

Table 3: Sensor interface Firewall Rulebase

Filtering for the scanner systems will be done on both interfaces. On the management interface, the rulebase to be installed on the management interface would look like this:

Source	Destination	Service	Action
Scanner – mgmt interface	Management Server	ntp, syslog	Allow
Management Server	Scanner – mgmt interface	ssh	Allow
Management Server	Scanner – mgmt interface	scanner management	Allow

Table 4: Scanner Firewall Rulebase - management interface

However, this rulebase can not be used on the scanning interface since this could have an impact on the scanning results of nessus and nmap. Therefore, a more selective filtering on the scanning interface is in its place. This can be done with the netfilter/iptables solution where filtering can be performed based on the flags in TCP Packets (and therefore on the state of these packets). Since we don't want anybody to connect to the scanning system through the non-management

interface, we have to stop “New Connections”. These new connections are detectable by the “SYN” flag (and only the SYN flag) in a tcp packet. In netfilter/iptables language, this is also known as a “state new” connection. The rulebase to be installed on the scanning interface would then look like this:

Source	Destination	State	Service	Action
Any	Scanner – scanning interface	New	Any	Drop

Table 5: Scanner Firewall Rulebase - scanning interface

Another general remark for the sensor and scanning systems is that they should definitely not have ip forwarding enabled.

4.4. Central management filtering

One can never be prudent enough, so additional filtering on the central management server is also a very good idea to implement. Like we’ve used on the sensor and scanner system, we can again use the netfilter/iptables firewall solution here.

The rulebase for the firewall on the central management server looks like this:

Source	Destination	Service	Action	Interface
Network Admin machine	Management Server	ssh, https	Allow	dmz
Snort Sensor	Management Server	ntp, syslog, mysql	Allow	mgmt
Scanner	Management Server	ntp, syslog	Allow	mgmt
Management Server	Snort Sensor	ssh, snort management	Allow	mgmt
Management Server	Scanner	ssh, scanner management	Allow	mgmt

Table 6: Central Mgmt Server Firewall Rulebase

Interface “dmz” is the interface connected towards the management DMZ. Interface “mgmt” is the interface connected towards the management communication network.

The first row in the table above shows additional filtering next to the main firewall filtering. This is needed to protect the central management server from hack attempts that are initiated from the management DMZ.

5. Components setup & deployment

5.1. General system security

Since all systems will have to be highly secured, the following basic guidelines should be taken into account in order to have a (very) secure scanning and detection network infrastructure.

The general security recommendations during the installation and configuration are the following:

- Only install packages that are really needed
- Disable services that are not needed
- Keep the system up to date

Keeping the system up to date could be performed by using an autoupdate script. This could have some security issues however that this way Trojans and backdoors could be introduced in the system. Only using the autoupdate script to download possible updates and checking the signatures of these updates before applying them is a better idea.

Since the iptables firewall solution will be used, the best thing to do is to recompile a new kernel. If a new kernel will be build, one can provide kernel security as well. This can be done through the grsecurity kernel patches which can be downloaded from http://www.grsecurity.net/. There are even modules for the current iptables tree to make the iptables firewall solution to behave like a stealthy firewall solution.

Additional guidelines are:

- The ssh daemon should only listen on port 22, protocol 2 only should be enabled, root access should be prohibited (to get access to the system, a regular account should be used instead of the root account) (see also 5.2.2)
- Prior in placing the system into the network, it should be checked with the "Linux Security Auditing Tool (lsat from http://usat.sourceforge.net/)". All errors and warnings should be investigated and corrected if needed.

5.2. Central management

5.2.1. General

The system needed for the central management server will have some hardware prerequisites. This because the following components will be running on this system: mysql database, apache webserver, acid snort log analysis, snort control center. For all these components together, sufficient memory, processing and hard disk space is needed.

The only services that should be reachable from the internal network (from the network administrator machine) are secure shell and https. Ssh for remote management of the management server (and other systems as well) and https for getting access to the acid central management console.

5.2.2. Secure Shell

The secure shell daemon that is installed on the management server should have some security settings modified in the configuration file (/etc/ssh/sshd_config). These are shown in the output below:

```
Port 22
Protocol 2
ListenAddress A.B.C.D
```

The secure shell daemon should only be listening on port 22 and should only allow protocol version 2. SSH protocol version 1 is vulnerable to several known security problems among which of them session hijacking. The ssh daemon should also only listen on the management dmz interface. This is done to prevent backwards connections coming from the sensors and scanners.

```
# Logging
SyslogFacility AUTH
LogLevel INFO
```

All authentication attempts should be logged to syslog.

```
# Authentication:
PermitRootLogin no
RhostsAuthentication no
IgnoreRhosts yes
PermitEmptyPasswords no
```

The root user should not be able to login remotely by any means. Logging in to this system has to be done with a regular user. If root access is needed, this can be obtained by issuing the "su –"- command.

```
X11Forwarding no
PrintMotd yes
UsePrivilegeSeparation yes

Banner /etc/ssh.banner

Subsystem       sftp    /usr/libexec/sftp-server
```

The central management server should not have Xwindow installed on it. This is only a recommendation. Xwindow can be installed if there is a need for it, if this is the case, X11Forwarding can be enabled so that the network administrator can launch X programs on his terminal in a secure way.

To adhere to legal issues, a warning banner (this is /etc/ssh.banner) should be configured and installed on the system stating that unauthorized access is prohibited and that logging and monitoring of all actions will be performed.

5.2.3. Mysql

The mysql database engine should have security applied to it as well. We don't want somebody to tamper with our intrusion detection logs and with our vulnerability assessment records do we.

- The mysql-user "root" has to have a password set. Setting a password for "root" can be done as follows:

  ```
  # mysqladmin -u root password <new password>
  ```

- The users that are used for data collection (either by the snort sensors or by the nessus scanners) should be restricted to be used from the host (ip address) of these sensors and scanners. For example: snort@sensor1 can only access the database "snort"
- These users used for data collection also have to have a password set, using the same method as for the "root" user.
- Don't give other users than the "root" user access to the user's database.

- Remove default users from the user's database.
- Mysql traffic should be encrypted. This can be done by tunneling the traffic through ssh or through a ssl tunnel. The ssl tunneling can be easily done by using the following option during the configuration and installation of mysql: "`--with-openssl`" (when compiled from sources only)

5.2.4. Apache

The apache webserver needs to be secured as well. Since sensitive data will be queried using the webserver, only requests through https should be allowed and access to the webserver on port 80 should be prohibited.

Changes in /etc/httpd/conf/httpd.conf to have a more secure webserver:

```
BindAddress <IP Address Management DMZ Interface>
#Port 80

<IfDefine HAVE_SSL>
#Listen 80
Listen 443
</IfDefine>
```

The guidelines shown above are for the default installation of the apache webserver on a RedHat based system. If the apache webserver is compiled and installed from the source files, then you will have to specify the certificates to be used by the server. The configuration files could then look like the following:

```
#Listen 80
Listen 443
SSLEnable
SSLCertificateFile /apache/conf/ssl.crt/server.crt
SSLCertificateKeyFile /apache/conf/ssl.key/server.key
SSLVerifyClient 0
SSLVerifyDepth 10
SSLLogFile /var/log/apache/SSL_log
```

An additional layer of security regarding the apache webserver is to password protect the server certificates. This password will be asked each time the apache web daemon is started.

```
<Directory />
#    Options FollowSymLinks
    Options None
    AuthName "Restricted Area"
    AuthType Basic
    AuthUserFile /usr/lib/apache/passwords/passwords
    Require user admin
    AllowOverride None
    Order deny,allow
    Deny from all
</Directory>
```

The network administrator should be prompted for a username and password before getting access to this webserver and its tools. Therefore, a password authentication mechanism has to be built in. This username/password combination is stored in an apache authentication file (see above). To create this password file

and to create the user “admin” at the same time, the system administrator would have to perform the following:

```
htpasswd -c /usr/lib/apache/passwords/passwords admin
```

An additional layer of intrusion detection and prevention can be build around the apache webserver by using the mod_security plugin.

Installation of the mod_security plugin on a RedHat based system is an easy task to perform:

```
# tar xvzf mod_security-1.5.1.tar.gz
# cd mod_security-1.5.1
# /usr/sbin/apxs -cia apache1/mod_security.c
# service httpd restart
```

The apache1 in the commands shown above resemble to the apache web server version 1.3.x.

The following has to be added to the httpd.conf (/etc/httpd/conf/httpd.conf) in order to be able to use the mod_security plugin. This is an example file that is included the mod_security distribution (httpd.conf.example-minimal)

```
<IfModule mod_security.c>

    # Turn the filtering engine On or Off
    SecFilterEngine On

    # Should mod_security inspect POST payloads
    SecFilterScanPOST On

    # Default action set
    SecFilterDefaultAction "deny,log,status:500"

</IfModule>
```

5.2.5. Acid

The acid console is accessed through the apache webserver with the mod_security plugin so additional security is not needed here. Sniffing of the acid remote management console is also not possible since the apache server is configured to only allow https traffic.

5.2.6. Nessus web interface

Inprotect (http://www.inprotect.com) has a web interface for nessus and nmap. The sad thing is that this is not a real distributed interface since it can only manage one single nessus daemon.

If a company has for example 2 or 3 nessus scanners (one in the DMZ, one in the internal network and one connected to the partner network), a simple solution would be to have 3 different instances of the inprotect web-interface, with 3 different inprotect-tables. This can be cumbersome to manage but has some advantages as well. The advantages are:

- Separate lists of vulnerabilities of the different scanned networks are possible

- Separated configurations are possible

Inprotect offers also the capability to launch nmap port scans (tcp, udp or both) towards a server to know what the open ports are on that server. The reports can be consulted through the web interface. These port scans are limited in the options that you can give to them. Only default scans are possible.

5.2.7. Nmap web interface

Nmap also has its own web interface, called nmapwebfe. This is again not a distributed but a local web interface. This is hardly usable in a distributed environment.

The solution to this problem can be solved by using rnmap (Remote nmap) which can be found on http://rnmap.sourceforge.net/. This tool consists of a server and a client application (console client as well as gui client). The connection between server and client is username/password protected and can be encrypted by means of ssl.

The first thing to do is to install and configure the server component (this has to be installed on the scanner):

```
# cp lib/*.* /usr/local/lib
# server/rnmapd
# server/rnmp-adduser
```

The last command is used to add a remote nmap user.

The second thing is to launch the client application. This can be done like the following example:

```
# client/rnmap -s <server-ip> -n "<nmap options>" <output options>
```

This can also be used in a script; the command then looks like this:

```
# client/rnmap -c <config file name> -n "<nmap options>"
```

The credentials and server ip address and port are stored in this file. Therefore the permissions on this file should be as secure as possible. A good permission setting would be "-r--------".

The nmapwebfe tool can be slightly modified so that it doesn't call the "real" nmap but that it calls the remote nmap client application. To use multiple port scanning engines, we can again use the simple solution of having 2 or 3 different webpage instances of the nmapwebfe web application.

5.3. IDS Sensor setup

The operating system installed should be hardened as much as possible. This includes among other guidelines the following:

- all unnecessary services are disabled
- all unnecessary packages are removed from the system (including development packages)
- all remaining packages are up to date

- The remaining services are configured as secure as possible (see also Ssh configuration earlier in this document).

The IDS Sensors network configuration has to be setup according to the following guidelines:

- The sensor interface should be set into promiscuous mode (with no ip address attached)
- The management interface should have an ip address set
- The management interface should have a firewall ruleset applied (see earlier in this document)
- IP forwarding should be disabled
- The ssh daemon on this system should only be listening on the management interface and only on port 22.

Note: Setting the interface into promiscuous mode could give problems sometimes, testing revealed that setting the interface to the ip address 0.0.0.1/0.0.0.0 actually did the trick.

All logging from these systems must be redirected to the central management station by using the syslog-ng facility.

Also here we can rebuild the kernel with the grsecurity security patches and latest iptables distribution included.

5.4. Scanner setup

The operating system installed should be hardened as much as possible. This includes among other guidelines the following:

- all unnecessary services are disabled
- all unnecessary packages are removed from the system (including development packages)
- all remaining packages are up to date
- The remaining services are configured as secure as possible (see also Ssh configuration earlier in this document).

The scanner network configuration has to be configured according to the following guidelines:

- The scanner interface should have an ip address in the network that it is scanning and should have the ruleset applied as described earlier in this document.
- The management interface should have an ip address set
- The management interface should have a firewall ruleset applied (see earlier in this document)

- IP forwarding should be disabled
- The ssh daemon on this system should only be listening on the management interface and only on port 22.
- The rnmap daemon should be running.
- The nessus daemon should only be bound on the management interface.

All logging from these systems must be redirected to the central management station by using the syslog facility.

Also here we can rebuild the kernel with the grsecurity security patches and latest iptables distribution included.

One single note though to remember when installing the scanner systems. The scanner systems need to have full access to the networks they are supposed to scan.

6. Analysis of results

Since all data is stored into a database, easy analysis and log correlation is possible. This by using the ACID centralized intrusion detection console for the intrusion detection systems and by using the web interface for the nessus scanners.

As of today, for analysing and comparing the results of nessus scans and nmap scans, there is no real “good” solution yet in the world of open source tools.

All syslog information from the different systems can be send to a mysql database as well. To be able to do this, all syslog daemons have to be replaced by the syslog-ng daemons. All these remote syslog-ng daemons have to be configured to send their data to the syslog-ng daemon running on the central management server. On this central server, the conversion from the syslog format towards mysql format is performed by using the method described on http://www.vermeer.org/syslog/. The syslog communication between the different systems can be tunnelled through ssl by using stunnel. Syslog data analysis can be performed by using own developed scripts or by using tools such as logcheck or others.

7. Conclusions

The big conclusion is that creating a secure solution for distributed intrusion detection and vulnerability scanning can be done with open source tools. However there are sometimes pitfalls and configuration difficulties that one has to concur in order to have a working solution. However, with a little effort, a good solution can be developed and deployed.

There also exist commercial tools for performing the different things mentioned here. For example, tenable security has also created a web interface infrastructure for nessus and nmap. This is called the Tenable Lightning Proxy (and Console). These are far more advanced than today’s publicly available tools but the cost can be significantly (especially for small companies with no (or none yet) security budget).

In order to have a secure distributed environment, one has to build in security measures in the manner of firewalls on the systems itself, additional service security settings (don't rely on default settings) and centralized logging.

8. References

[1] Law, Seth. "Low cost network security". Sans Institute. 31 May 2003. URL: http://www.giac.org/practical/GSEC/Seth_Law_GSEC.pdf (8 August 2003)

[2] May, Artur. "Securing Apache: Step-by-step". Security Focus. 14 May 2003. URL: http://www.securityfocus.com/infocus/1694 (9 August 2003)

[3] Dens, Stefan. "SnortCenter & SnortCenter agent installation". 2002. URL: http://users.pandora.be/larc/documentation/chap1.html (9 August 2003)

[4] Danyliw, Roman. "Analysis Console for Intrusion Databases (ACID)". URL: http://www.andrew.cmu.edu/~rdanyliw/snort/snortacid.html (9 August 2003)

[5] Scott, Steven J. "Snort Installation Manual - Snort, MySQL, Redhat 7.3" URL: http://www.snort.org/docs/snort-rh7-mysql-ACID-1-5.pdf (8 August 2003)

[6] Huber, Robert. "Strategies for improving Vulnerability Assessment Effectiveness in Large Organizations". Sans Institute. 31 May 2003. URL: http://www.giac.org/practical/GSEC/Robert_Huber_GSEC.pdf (8 August 2003)

[7] "General Security Guidelines". www.mySQL.com. URL: http://www.mysql.com/doc/en/General_security.html (9 August 2003)

[8] Ristic, Ivan. "Mod_Security Reference Manual". 10 July 2003. www.modsecurity.org. URL: http://www.modsecurity.org/documentation/modsecurity-manual-v1.5.1.pdf (9 August 2003)

[9] Alex. "Configuration guide for inprotect". 14 July 2003. URL: http://www.inprotect.com/modules.php?op=modload&name=Diner_Wrapper&file=index&req=ShowFile&file_wrap=html/readme.html (10 August 2003)

[10] Westphal, Kristy. "Secure MySQL Database Design". SecurityFocus. 18 February 2003. URL: http://www.securityfocus.com/infocus/1667 (16 August 2003)

[11] Scheidler, Balázs. “Syslog-ng reference manual”. 2000. URL: http://www.balabit.com/products/syslog_ng/reference/book1.html (16 August 2003)

[12] “Centralized syslog-ng to mysql database”. www.vermeer.org 18 May 2002. URL: http://vermeer.org/syslog/ (16 August 2003)