Building a Security Management Point

by Flavio Marcelo Amaral

last updated Thursday, April 14, 2000

Introduction

Keeping networked environment secure can be a very difficult task nowadays. There are many means of attack. One must always be aware of the vulnerabilities discovered most recently, not to mention the eternal activity of viewing logs and looking for suspicious traces. All this takes time and becomes worse when we are dealing with networks. Watching a single machine requires careful effort, doing the same with an entire network may take many times the effort. What we need is to increase our watching range with as little time as possible.

A well implemented security environment can make the most of the security manager's time by allowing him to do carry out his/her main tasks with less effort. This article will present a way to build a security management point by using free IDS solutions to watch the following:

- Signs of attacks in the hosts system logs
- Changes in important files/directories used to hide an attack
- Stations on the network that have been a target of a port scan
- Login accounts that have been compromised
- User login habits

By implementing tools for detecting and reporting this data to a single management point, we can save a lot of time by watching what is going on all workstations.

Reducing time to read logs

Reading logs can be a very tedious and error prone activity. Attacks may pass undetected when it is mixed with thousands of lines of normal activities. What we need is an Intrusion Detection tool to analyze logs for us and extract only relevant information. There are tools such as logscanner and logcheck that search syslog looking for suspicious actions.

Logcheck is a component of the Abacus Project which processes logs generated by other Abacus Project tools, system daemons, TCP-Wrapper, logdaemon, and TIS Firewall toolkit. It goes straight to the point by extracting important data from violations detected by these tools.

Logcheck does not operate constantly and the system administrator must configure crontab to run it at certain intervals. It utilizes a program called logtail that records the last position read from inside the log file and begins from this position when it is run again. It works by comparing its database of attack signatures with the events logged. There is also a database of keywords that indicate false positives, therefore one can place there what one desires to be ignored. This makes the program very flexible, allowing it to adapt itself to many different environments.

It can be installed on most Unix machines on a network and watch for intrusion attempts in the services offered by each one. For example, one may configure it to catch sendmail abuses, who are trying to guess passwords by using pop3, etc.

Logcheck reports

Logcheck reports are sent to an e-mail account, allowing the tool to be installed on many workstations and reports are sent to a central location. Below is an example of a message sent by logcheck. The messages can have three different headers: active system attack alerts, security

violations and unusual system events.

Active system attack alerts

```
Jan 30 00:27:36 somehost portsentry[32105]: attackalert: Connect from host:
foo.bar.com/10.1.1.52 to TCP port: 5742
```

Security violations

```
Dec 21 22:00:55 somehost ipop3d[22419]: Login failure user=joe host=
[10.1.1.115]
Dec 22 01:09:30 somehost sendmail[27738]: BAA27738: ruleset=check_rcpt, arg1=,
relay=[200.241.99.247], reject=550 ... Relaying denied
```

As we can see, logcheck works very well with other Abacus project tools by highlighting their reports (we will discuss PortSentry later). It is also a very flexible tool and can suit many networks. Its attack signatures database can be very easily updated and adapted to all hosts on a network. It is a very fast way to be informed about what is going on, on all hosts.

Checking the integrity of systems

The new trend of DDOS attacks had caused many systems to be compromised and participate in attack sessions. It is imperative that a system administrator knows that the systems are clean and no unauthorized changes were made.

One of the first things an attacker does is replace certain system utilities to hide their presence. If we want to detect those changes, we must install a file integrity checker. This type of program watches files and directories and notifies the system administrator of any changes.

AIDE (Advanced Intrusion Detection Environment) is a tool to check the integrity of files and directories. AIDE creates a database from regular expression rules it finds from the config file. It works by checking the file attributes and has 4 message digest algorithms (md5, sha 1, rmd 160 and tiger).

AIDE can be installed on all supported Unix machines on a network and set up accordingly to protect their file systems. It is very important that besides taking care of system binaries (ifconfig, ps, netstat etc), directories that are seldom modified (/usr/man, /usr/doc etc) are also under protection. They are good places to hide things since few people go there. The system administrator can configure crontab to run AIDE at any number of times a day on every machine or can use a floppy if he doesn't trust the host or wants to check it again.

AIDE reports

Below are some lines of a typical AIDE report.

```
Summary:
Total number of files=13844,added files=619,removed files=146,changed files=138
removed:/etc/prog.conf
changed:/etc/passwd
added:/usr/man/man1/....
```

Since AIDE reports its findings to a file, it can be useful to configure a script to e-mail the file to the account of the system administrator right after the check. If we do this on every machine, we can watch all the file systems from a single position.

Detecting and blocking port scanners

Networking port scanners are often used before an attack since it can reveal valuable information to the attacker such as: version of the OS, what services are currently running, if there is any service offering an open door, etc. With this information, the intruder can consult a database of exploits and may launch a successful attack.

Some port scanners may leave footprints and be caught by syslog. Others do the job very quietly and it is necessary to possess an appropriate mechanism to catch them. There are many tools designed to detect port scans at the moment. Some go even further by blocking connections from the hostile host in real time.

PortSentry is a scan detector that blocks connections by using /etc/hosts.deny file. Its main features are:

- Extensive stealth scan detector support for: FIN, half open, NULL, "oddball packet", SYN and X-MAS style attacks and UDP scans
- Simultaneous UDP and TCP monitoring of multiple sockets

To detect scans, PortSentry has four stealth scan detection modes:

1. stcp stealth tcp scan detection mode
2. sudp stealth udp scan detection mode
3. atcp advanced tcp stealth scan detection mode
4. audp advanced udp stealth scan detection mode

Modes 1 and 2 operate by watching a list of ports configured by the user. Modes 3 and 4 are the most sensitive modes because they cover a wider range of ports (the default is any port below 1024). The system administrator must be very careful with the advanced modes as it may result in high sensitivity and cause a great influx of false positives. Since an attacker may spoof IP source addresses, PortSentry may block traffic from a host which should not be blocked. Therefore, there is an option in the configuration file named IGNORE_FILE pointing to a file that contains IP addresses from hosts that shall never be blocked. This automatic blocking resource can be disabled as well.

When a port scan is detected, PortSentry can react in the following ways:

- logging the incident in syslog
- putting the attacker IP address in /etc/hosts.deny
- dropping the connection through configuration of route table and packet filters

Three types of messages are sent by PortSentry to syslog

- admin alert - operational status
- security alert - a security relevant event has occurred
- attack alert - a port scan attempt

Active system attack alerts

The following shows a logcheck report. We can see how it operates together with PortSentry to report scans:

```
Jan 30 00:27:36 patrol portsentry[32105]: attackalert: Connect from host:
foo.bar.com/10.1.1.52 to TCP port: 5742
Jan 30 00:27:36 patrol portsentry[32105]: attackalert: Host 10.1.1.52 has been
blocked via wrappers with string: "ALL: 10.1.1.52"
Jan 30 00:27:49 patrol portsentry[32105]: attackalert: Connect from host:
```

```
foo.bar.com/10.1.1.52 to TCP port: 1080
Jan 30 00:27:49 patrol portsentry[32105]: attackalert: Host: 10.1.1.52 is
already blocked. Ignoring
```

All of the alarms are intercepted by logcheck and e-mailed to the system administrator. This allows him/her to be informed about all scan attempts on a network.

Scans coming from internal hosts can be detected in advanced mode with a low error rate. If it is a false positive, the impact will be small. On the other hand, if this impact were on hosts directly connected to the outside world, it could disturb the normal communication. Therefore, the author recommends that advanced mode be installed in intranet environments and normal mode in hosts in a demilitarized zone.

Checking login accounts activities

In a network with many different users, a system administrator may want to know if users are using the systems is according to the site security policy and if there is a compromised login account.

HostSentry is an Intrusion Detection tool designed to detect login anomalies by alarming when a user does something which is very suspicious with their account. It watches wtmp in real time by building a separate audit trail about login and logout information. Controlling who is logging onto the network, and the way it is done, can be useful to know at an early stage to see if an inside user is planning an attack, or if the account was compromised and is now in the hands of a cracker. The program has a Login Anomaly Detector (LAD) mechanism that reports the following:

- bizarre behavior: a user is doing something very suspicious
- time anomalies: when users log in at unusual time
- local anomalies: when connections are made from places never seen before

The program classifies anomalies as:

- user has created a .rhosts file containing "++"
- user has created a directory named "..." , ".. " etc
- user has erased his history file
- user is logging in from a strange origin

HostSentry has only been tested on Linux and OpenBSD and requires python to run. The program will take time to learn normal login and logout patterns and user behavior. Once this phase is done, a system administrator will have much better control of all user accounts and accesses.

HostSentry reports

The following shows a warning about a login from a foreign domain:

```
Jan 30 00:27:49 patrol hostsentry[32405]: securityalert: foreign domain login
detected for user:joe from: 10.1.1.1
```

This tool also works with logcheck.

Conclusion

With these tools properly set up and running, one can protect an entire network without so much pain. They offer a fast mechanism to warn a system administrator about intrusions. Besides their IDS resources, one of their main strengths is that they are all highly configurable. It's impressive how we can adapt them to our reality.

A security manager can prepare a workstation to serve as the central security management point of a network. Needless to say that this host must be configured with only security in mind. From this point, they can visualize logs more quickly, watch scans, check file system integrity and all user behavior. Protecting all these levels is good security practice since a lot of data can be collected from the workstations. If any of them is attacked, this information will be at hand and may lead to the identification of the attacker.

Flavio Amaral is a network security engineer. He is currently completing his Masters Degree in the security field (IDS). He has been working in the security field for 5 years, setting up security tools and monitoring networks. Flavio holds a degree in computer science and 5 years experience in Unix system administration.