Global Information Assurance Certification Paper

Auditing a Linux Honeyd Honeypot

Auditing Networks, Perimeters, and Systems

GSNA Practical Assignment

Version 4.0

Option #1

Author: Hugo Cira

Date: 19 March 2005

Attended at Las Vegas, NE

Octuber, 2004

1. Abstract

At the end of the 90's and during this decade, the Information Security world has grown exponentially, and the organizations have learned painfully that they have to protect their information assets from a high variety of kind of attacks that grow every year.

Among the universe of security tools that have been developed to protect our networks, as Firewalls, IDS (Intrusion Detection Systems), IPS (Intrusion Prevention Systems), etc., there is a relative new kind of security tool called Honeypot. Spitzner, L. (2003) defines a Honeypot as follows:

"A Honeypot is a security resource whose value lies in being

probed, attacked, or compromised".

This security resource is so flexible, that the organization can use it to detect intrusion, unethical behavior of employees, delay attack to their networks, forensics, gather information of attacks to know how they were made, virus researching, etc.

In this document we are going to audit a Honeypot, deployed in a Linux box. Remember that the goal of a Honeypot is being probed or attacked, so this is not a regular audit. There are services and ports that should be opened proposing to allow the attackers to get in. But in other way, we have to assure that once this Honeypot is accessed, it won't be used to compromise other systems.

We are going to see a very short review about Honeypots, then we are going to specify the organization objectives, describe the system itself that we are going to audit, identify risks associated with the system and their implications, develop tests to detect the presence of the vulnerabilities that lead to those risks, and finally, we are going to run those tests and generate and report the results.

Table of Contents

List of Figures

List of Tables

2. The System

Among the universe of security tools that have been developed to protect our networks, as Firewalls, IDS (Intrusion Detection Systems), IPS (Intrusion Prevention Systems), etc., there is a relative new kind of security tool called Honeypots. Spitzner, L. (2003) defines a Honeypot as follows:

> *"A Honeypot is a security resource whose value lies in being probed, attacked, or compromised".*

Basically, a Honeypot is a system that simulates production systems, for instance, HTTP servers, SMTP servers, a whole network, etc. The idea is to gain the attention of attackers that, intentionally or not, want to get in to your network. Once they can have access to your network, the Honeypot has to have mechanisms to log every activity, prevent the attacker to get access to other systems (internal or external) and disconnect the attacker if the things go out of control.

The Honey Net Project (The Honey Net Project, 2004) have deployed Honeypots around the world with the objective of learning more about those techniques used by Hackers and help the community to develop tools and procedures to prevent attacks and detect them quickly to reduce the impact in our organization.

Researching is the main goal of Honey Net Project. However, there are other organizations that want to use Honeypots as a simply Intrusion Detection System. They don't care about how hackers get into their networks, just want to have an effective mechanism to detect them when they break in. Other organizations would like to deploy a Honeypot as a vigilance resource in order to detect any employee behavior that can be out of the ethical line. Other organizations can use this tool to simulate production environment and systems and detect possible sources of fraud.

As you can see, this security resource can be used in many different ways to reach different goals. That is why we have to define very clearly, the objectives of an organization that decides to deploy a Honeypot in its network.

2.1. Organization Objectives

1. The organization wants to deploy a Honeypot whose role will be detecting unauthorized activity in the corporate network. The organization plans to deploy this security resource as a complement of the IDS deployed in the perimeter of the corporate network. The IDS only reports events analyzing the traffic that passes through the perimeter. When a machine starts to mess up the internal network due to a virus installed on it or an intentional attack, the IDS will never detect this activity unless this activity goes to external networks.

2. The Honeypot must collect enough information to identify the source of the attack and the tasks performed during the incident by the attacker. With this information, the organization will be able to check the source locally in order to identify what was the cause of the problem and mitigate the risk.

3. The organization is planning to use Honeypots in the future to detect fraud in some critical systems. For this reason, they are going to start with a simple and low risk Honeypot to gain experience and gather enough information to deploy more complex Honeypots in the future.

In order to achieve these objectives, the organization decided to deploy a low-interaction Honeypot. According to Spitzner (2003), the Honeypot can be classified according to the level of interaction that the attacker can have with it. In other words, the more the Honeypot seems to be a real system, the more interaction the attacker will have. For instance, a low-interaction Honeypot can be a port listener program that logs any connection made without actually performing the tasks of the real service. The attacker is not allowed to do anything but open and close a connection; that is why this is a low-interaction Honeypot. On the other hand, a high-interaction Honeypot can be a server with real services running, with mechanisms to control the actions the attacker can do inside, and logging the activity of the attacker inside the Honeypot.

From the last example, we can deduce that high-interaction Honeypots are riskier. A low-interaction Honeypot will allow the organization to detect unauthorized activity in the network it is placed. Because this is not a production

system, it is not supposed to receive any connection from real users, so any connection attempt against the Honeypot is considered suspicious. Also, the Honeypot is easy to deploy and offers low risk. So a low-interaction Honeypot seems to match the organization's objectives.

Now let's get down into business and talk about the system we are going to audit. In the diagram bellow we can see the location of Honeypot in the network.

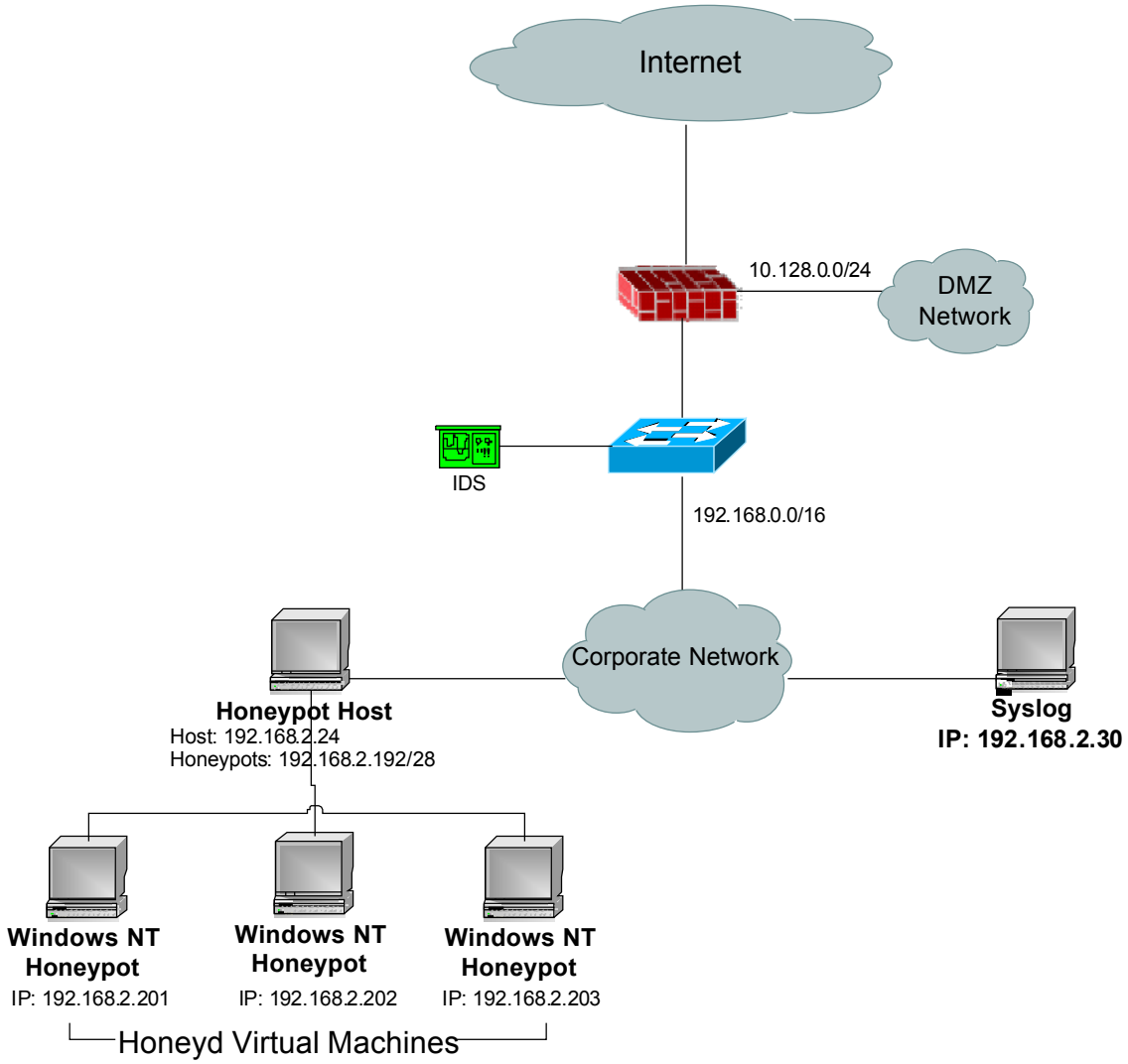

Figure 1: Honeypot's logical location inside corporate network

The addresses 192.168.x.x shown here are not real in order to protect the identity of the organization. These IP addresses are valid and can be reached from the internet; there is no Network Address Translation (NAT) in the corporate network. The following table details the hardware and software specification of the Honeypot:

Hardware	Server HP DL320
Processor	P4 2.26 GHz
Mem (RAM)	1GB
Hard Disk	IDE ATA 80GB
NIC	2 x Gigabit
Operating System	Linux Fedora Core 3
Honeypot Software	Honeyd 1.0 / Arpd
Data Capture Software	TCP Dump
Intrusión Detection System	Snort 2.3
Data Control Software	IPTables

Table 1: Honeypot specifications

In this paper we are going to name *The Host* to the system that has the Honeypot software installed on it. In this case, the host will be the Linux Red Hat System. Honeyd, of which we will talk in the next section, will be the Honeypot Software installed in the Host. Also we are going to use Honeypot and Virtual Machines making reference to the devices generated by Honeyd.

2.2. Honeyd

We are going to start with the main piece of software of the System, the Honeypot system. Honeyd is an open source Honeypot system developed by Niels Provos (http://www.honeyd.org/), originally for Unix systems, but with the time there have came up versions for Windows Platforms.

Honeyd is one of the best and popular Low-interaction Honeypots that we can find in the world of the Honeypots systems. Let's take a look at its features

(Spitzner, 2003):

- It can emulate different services in a variety of devices out of the box: Windows NT, Windows 2000, Linux, Solaris, Cisco, etc.

- It emulates services not only at application level but at TCP/IP stack. It makes it more difficult to detect by fingerprinting with tools like nmap or x-probe.

- It can be personalized to emulate other services besides the standard that comes with it. For instance, a script can be developed to listen in any port and interact with the attacker to make it more real and collect more information about the attacker.

- Unlike the majority of Honeypots solutions, Honeyd can listen on any unused IP in the network, assume their identity and then interact with the attacker. This is done with a complement tool called Arpd, installed along with Honeyd.

- Logging capabilities. Honed can log any TCP, UDP or ISMP activity. More logging can be added in the scripts that emulate services.

- Easy to configure and deploy. This can be deduced from the points above. You can have a variety of devices just installing and adding a few lines to a configuration file.

- It's free

For a quick deployment, we used a Honeyd Toolkit (http://www.tracking-hackers.com/solutions/). This toolkit contains all the configuration files, precompiled static binaries, and startup scripts to get Honeyd instantly up and running in a Linux computer. We also used some scripts to emulate additional services as exchange IMAP and SMTP. These scripts can be found in http://www.honeyd.org/contrib.php.

The Honeypot was configured with 3 virtual systems: Windows NT, Linux and Cisco router. Bellow we can see the Honeyd configuration file:

```
### Windows computers
create windows5
set windows personality "Windows NT 4.0 Server SP5-SP6"
set windows default tcp action reset
set windows default udp action reset
add windows tcp port 80 "perl scripts/iis-0.95/iisemul8.pl"
add windows tcp port 139 open
add windows tcp port 137 open
add windows udp port 137 open
add windows udp port 135 open
set windows uptime 54298453
bind 192.168.2.201 windows

### Linux 2.4.x computer
create linux
set linux personality "Linux 2.4.16 - 2.4.18"
set linux default tcp action reset
set linux default udp action reset
add linux tcp port 110 "sh scripts/pop/emulate-pop3.sh"
add linux tcp port 25 "sh scripts/smtp.sh"
add linux tcp port 21 "sh scripts/ftp.sh"
set linux uptime 12542494
bind 192.168.2.202 linux

### Cisco router
create router
set router personality "Cisco IOS 11.3 - 12.0(11)"
set router default tcp action reset
set router default udp action reset
add router tcp port 23 "/usr/bin/perl scripts/router-telnet.pl"
set router uid 32767 gid 32767
set router uptime 16876456
bind 192.168.2.203 router
```

Figure 2: Honeyd Configuration File (based on Spitzner, March 12, 2003)

In the following table we can see a summary of the services running by virtual machine:

Honeypot IP	***Port***	***Service***
192.168.2.201	80 TCP	http
	139 TCP	NETBIOS Session Service

	137 TCP	NETBIOS Name Service
	137 UDP	NETBIOS Name Service
	135 UDP	DCE endpoint resolution
192.168.2.202	110 TCP	Pop
	25 TCP	smtp
	21 TCP	ftp
192.168.2.203	23 TCP	Telnet

Table 2: Virtual Honeypots and services

Now we are going to see components to capture the data of the connections made against the Honeypot and the mechanisms to control those connections.

2.3. Data Capture

The Host has also complement mechanisms for data capture in order to get more evidence of the attacks:

2.3.1. Snort

Snort (http://www.snort.org) is an open source software, designed to be a lightweight network intrusion detection system, capable of performing real-time traffic analysis and packet logging. Snort has been deployed in this solution, as a Intrusion Detection System (IDS), to detect any malicious activity against the Honeypot and trigger alarms. The snort alerts are sent to a Syslog server.

The configuration of snort for this solution is basic. A couple of lines and variables of the default snort.conf were modified. The most important changes are:

`output alert_syslog: LOG_AUTH LOG_ALERT`. Uncomment this line so that the snort Alerts are sent to the syslog server

`log ip any any <> any any (msg: "Snort Unmatched"; session: printable;)`. This rule is set to log everything to a text file. This is good because it logs the whole sessions in a clear text, which is helpful for post-analysis with automated scripts.

Everything else is set in the command line, when finally snort is launched. Here is the command:

```
snort -i eth0 -de -h 192.168.2.192/28,192.168.2.24/32 -l
/var/log/snort/ -c /usr/local/src/snort-2.3.0/etc/snort.conf -D
```

2.3.2. TCPDump

The other goal is to capture all the traffic inbound and outbound the Honeypot. Although this can be done with Snort, with TCPDump we have more flexibility and more options. This is the command to launch TCPDump to capture traffic in the host:

```
/usr/sbin/tcpdump -i eth0 -n -s 1500 -C 2 -w
/var/log/tcpdump/tcpdump.log
```

With the option –w will allow us to open the log with a traffic analyzer tool as Ethereal. We will see this in more detail ahead in this document.

2.3.3. System Logs

System logs. All the events registered by the operating system are sent to the syslog server.

2.4. Data Control

Other issue that is required in a Honeypot deployment is a data control mechanism. The goal of data control is to keep track of the attacker activity against the Honeypot and from the there to other hosts in internal or external networks. It is very important that we don't let the attacker to harm other

systems if he takes control of the Honeypot. The Honeypot system counts with the following mechanisms for data control:

2.4.1. Firewall Filters

The firewall is the first layer of defense located in the perimeter of the corporate Network (see Figure 1: Honeypot's logical location inside corporate network). Here we can control inbound and outbound connections from and to external networks. Although this device protects the Honeypot from being harmed from outsiders, the main objective of this layer of data control is to prevent the Honeypot being used to attack external systems.

According with the organization objectives, we need to detect malicious activity inside the corporate network, this activity may comes from outside networks. The inbound-firewall-rules will be the same that applies to all corporate networks, this way, the Honeypot will be a tool to detect any security breach in the perimeter.

We also are not going to allow any connection from the Honeypot to outside networks. Since any connection made to the Honeypot is suspicious, the Honeypot already achieved its goal detecting them and logging the information, necessary to identify the source.

2.4.2. IPTables

IPTables is a framework inside the Linux kernel that provides packet filtering, network address translation (NAT), among other functions related with packet management.

The IPTables are used as a protection at Host level. It would be the same functionality than Firewall rules but controlling inbound and outbound connections between corporate network, and the Honeypot itself. As it was mentioned above, we are allowing any inbound connection to the Honeypots and not allowing outbound connections.

Also IP Tables are configured to assure that no connections are made against

the Host IP unless it is administration-purposes traffic like SSH, etc. All this sentences are mapped in the following IPTables commands:

```
$IPTABLES="/sbin/iptables"
# Allow the following inbound inbound
$IPTABLES -A INPUT -p tcp --tcp-flags ALL SYN --dport 22 -j LOG --log-
prefix 'Inbound SSH Connection: '
$IPTABLES -A INPUT -p tcp --tcp-flags ALL SYN –d 192.168.2.24  --dport 22 -j ACCEPT

$IPTABLES -A INPUT -d 192.168.2.201 -j ACCEPT
$IPTABLES -A INPUT -d 192.168.2.202 -j ACCEPT
$IPTABLES -A INPUT -d 192.168.2.203 -j ACCEPT

# Maintain state of inbound connections.
$IPTABLES -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT

### Set Policies
$IPTABLES -P INPUT DROP
$IPTABLES -P FORWARD DROP
$IPTABLES -P OUTPUT ACCEPT
```

Figure 3: IPTables configuration commands (based on Spitzner, March 12, 2003).

The INPUT filters related with the Host assure that no connection but SSH with the SYN flag turn on will be blocked. This also means that any connection open from the Host to outside will not succeed because the answer will be blocked.

2.5. The Scope

Although we have described the whole Honeypot solution deployed, we are going to focus this analysis in the Honeypot box, which includes The Host and the 3 virtual Honeypots configured with Honeyd, along with the components for Data Capture and Data Control configured on the box. We are not going to study vulnerabilities in the other external components of the solution: Firewall or Syslog server, directly. This means that we are going to assume that they are very well secured and the level of risk and exposure of those components is low.

3. Risks, Impacts, Vulnerabilities and Exposure

As we have said at the beginning, the risk level of having a Honeypot in a network increases as the level of interaction is higher. In this solution the

organization picked a low-interaction Honeypot because the functional objectives were met with a low risk. However, this does not mean that there is not risk at all. We are connecting a server that can be reachable from inside and outside networks, so that if this server is not secured it can be hijacked and used to harm other systems. What we are going to see in this section is to analyze the main risks of having this Honeypot connected in the organization's network.

3.1. Risks

There are three main risks that we have to be careful with in this system:

1. The Honeypot can be hijacked and used to attack internal networks

2. The Honeypot can be hijacked and used to attack external networks.

3. The Honeypot can be hijacked silently by an external attacker and used to sniff traffic in the corporate network. Also, the attacker can scan internal networks to have more information about topology, vulnerabilities, etc.

Although some measures were taken above to decrease the risk and impact: IPTables, snort, tcpdump, syslogs, etc., those risks are always present because we are hoping that somebody probes our Honeypot. If our Honeypot is extremely secured and never gets a connection, we are not doing our job.

3.2. Impacts

The impacts, if the risks we listed above are met, are important. Here are the main ones:

1. If an attacker takes control of the Honeypot and attacks systems in the internal network, it can cause damage with different ranges of impacts. For instance, if the attacker can only scan the network and harm a few desktops, it would be a low level impact. Otherwise, if the attacker can harm a platform that hosts a service for customers, for instance a Web

Hosting, email platform, portal, etc., the impact is high because this affects external customers operation, there is money lost, and the image of the company is affected.

2. If an attacker takes control of the Honeypot and attacks external networks, there are a lot of consequences that we can list. Our organization might be placed in a blacklist or blocked in the Internet.

3. Legal issues can be involved if the previous attacks happen because the organization is facilitating the attack and is failing to take steps to prevent the use of the system. Also, if the attacker opens an IRC channel and this communication is intercepted, the organization can be sued because this is an illegal Wiretap Act (SANS Institute, 2004).

4. By sniffing internal traffic, the attacker can steal critical information for the organization like, email users and passwords, marketing plans, strategic plans, etc. Also, the attacker can scan the corporate network to get useful information about topology, vulnerabilities, etc., and this information can be used to perform other attacks from external networks. The attacker can be nasty enough to public this information in a web site so that other hackers can do their own attacks.

3.3. Vulnerabilities

There are a lot of vulnerabilities that can be exploited and make those risks happen. We are going to specify the main vulnerabilities that we could find in The Host and Honeypots.

3.3.1. Host Vulnerabilities

We call The Host to the operating system where the Honeypot is installed in. For vulnerability analysis proposes, we are going to include as a part of The Host: the Operating System, local Data Capture mechanisms (Snort, system logs, syslogs, etc.) and local Data Control (IPTables configuration). The following are the main vulnerabilities that can lead to the risks we mentioned above:

1. Operating System unpatched.

2. Lack of security configuration at the operating system level. In this point we can include these major vulnerabilities:

   2.1. Users (e.g. root) with blank or weak password.

   2.2. System files permissions (e.g. `/etc/password`, `/etc/shadow`, start-up files, libraries, binaries, etc.).

   2.3. Services running with excessive permissions. There are vulnerabilities in some applications or services that, if they are exploited, the attacker can get access to the system with the user configured for running that server. So it is a good practice to avoid running services with root privileges.

   2.4. Lack of security events logging. It is important that the system is configured properly to log security events and any other operation or access to critical files of the system.

   2.5. Unnecessary services, started and vulnerable.

   2.6. Lack of control of inbound/outbound connections to hosts located in internal or external networks.

3. Software unpatched installed: in our case, we have to check patches for snort, TCPDump and IPTables.

3.3.2. Honeypot Vulnerabilities

The Honeypot will be the software installed to generate the Honeypots (Honeyd) and the Honeypots themselves. Depending in the services we simulate, we can have more or less vulnerabilities. Here are the major ones:

1. Simulated services that can allow the attacker to open connections to other systems (e.g. an open relay SMTP service)

2. Simulated services that allow the attacker to access local resources of the Honeypot. (e.g. FTP service that allow access to the real filesystem).

3. Scripts that can bring down the system with a denial of service or a bug.

4. Scripts downloaded from public sites that may have backdoors.

3.4. Scenario of exposure

As we can see, these risks can be very common and be present in any server that we connect in our network. The thing is, at least those servers are connected because they are really necessary, they are giving a service and without them there would not be a business.

Also, when a production server is connected in the network, and It hosts services like a portal that must be accessible from external networks, the administrators take all the secure points to assure that only the services needed can be let go through. The server is placed in a DMZ, outside the corporate network, etc. In our case, there is more exposure because we are having this element inside the network with some services open in order to be probed or attacked. Remember that this is the main goal of a Honeypot.

In this solution, the exposure is not that much since the Honeypot is not 100% accessible from the Internet. There are not special rules in the perimeter firewall for inbound connections against the Honeypot though, just the corporate filters. Since the Honeypot is configured inside the corporate network, this is a good way to verify and detect if the Firewall is not doing its job.

Also, due that there is a low-interaction Honeypot that just lets the attacker to interact with simulated services, the exposure is even lower.

4. Testing and Reporting

In this section we are going to design a set of tests that will allow us to identify the presence or not of the vulnerabilities listed above in the system. Since the main vulnerabilities are related to the Host, our tests will be focused mainly to catch vulnerabilities at the host side. There are other tests related with Host and Honeypots and the last one with the Honeypot.

4.1. Checklist

The first step we want to take is to perform a check over the basic security points according to the best practices. The Center for Internet Security (http://www.cisecurity.org) has developed a series of security benchmark tools that automates their checklist with Perl scripts. There is a benchmark scoring tool (http://www.cisecurity.org/bench_linux.html) for Linux Red Hat which applies to our Linux Fedora Core 3 installation. The nice thing about this tool is that we can get a score that tells us how good or bad we are, which is good for having a criteria of passing or failing the test.

The first thing we have to do is download the tool from http://www.cisecurity.org/bench_linux.html. Then, uncompress the file and execute it as follows:

```
[root@rs26s6 CIS]# ./cis-scan
```

In the Appendices CIS Benchmark Scoring Tool Results you will see the output of the command. The tool generates a log file with every point it checks and the results. This log is the most valuable thing of this test, because it gives the Administrators specific configuration directives to secure the system. In the same Appendices, there are two lists of the Positives and Negatives points the tool found.

The tool calculates a score based on the balance between positive and negative points. Here is our score:

```
Ending run at time: Thu Mar 17 19:16:19 2005
```

```
Final rating = 6.00 / 10.00
```

Figure 4: cis-scan results

According to the tool, a good score would be between 6.00 and 10.00. According to my experience with this tool, if we want to have a server well secure, we are going to need an 8 or more. You can see it in the Appendices, the list of Negative point is bigger than Positive's. So this is a point that needs more work by the Administrators.

4.2. Port Scanning

Port scanning and vulnerability assessment is another basic step that we have to take in order to detect vulnerabilities in the Host. Also, we will find very nice things scanning the Honeypots.

4.2.1. Netstat

Our first test is to look at the open ports from inside the Host. This will get us an assessment of what services are running in the server and detect any unnecessary server opened.

Netstat is a tool shipped in the Operating System, so we don't have to install anything, just run the command as follows:

```
 [root@rs26s6 tools]# netstat -a -p --inet
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address               Foreign Address             State       PID/Program
name
tcp        0      0 *:32768                     *:*                         LISTEN      4493/rpc.statd
tcp        0      0 *:sunrpc                    *:*                         LISTEN      4473/portmap
tcp        0      0 localhost.localdomain:ipp   *:*                         LISTEN      4691/cupsd
tcp        0      0 localhost.localdomain:5335  *:*                         LISTEN
4657/mDNSResponder
tcp        0      0 localhost.localdomain:smtp  *:*                         LISTEN      4879/sendmail: acce
udp        0      0 *:32768                     *:*                          4493/rpc.statd
udp        0      0 *:bootpc                    *:*                          4175/dhclient
udp        0      0 *:853                       *:*                          4493/rpc.statd
udp        0      0 *:5353                      *:*                          4657/mDNSResponder
udp        0      0 *:sunrpc                    *:*                          4473/portmap
```

```
udp        0      0 *:ipp                   *:*                                 4691/cupsd
raw        0      0 *:255                   *:*                    7            7166/honeyd
```

Figure 5: Netstat command output

Here we found an unnecessary and very dangerous service opened: Sendmail. If this service is not configured it will allow any person to send emails anywhere, which will be an Open Relay. If this port is not closed in the perimeter firewall, spammers can use this server to send email inside and outside the organization. So this can bring serious consequences to the organization.

4.2.2. Nmap

Nmap is a port scanner tool that has become one of the most popular in the security community. Nmap can be downloaded from www.insecure.org/nmap/. One of the features in which we are more interested here is the Operating System detection by fingerprinting feature of Nmap. This will allow us to test if the virtual operating systems are well simulated.

In the last test, we looked at the ports open from inside, which was very important to detect unnecessary services opened. We are going to execute this test from a remote machine 192.168.2.34. This will allow us to see how the Host and the Honeypots are seen from the corporate network and test how IPTables are working.

First, we are going to start with The Host: 192.168.2.24. Here is the command and the results:

```
C:\Program Files\nmap-3.81>nmap -sS –P0 –O 192.168.2.24

Starting nmap 3.81 ( http://www.insecure.org/nmap ) at 2005-03-17 19:05 SA Western Standard
Time
Interesting ports on hostname (192.168.2.24):
(The 1661 ports scanned but not shown below are in state: closed)
PORT    STATE SERVICE
22/tcp  open  ssh
111/tcp open  rpcbind
MAC Address: 00:0E:9B:28:B6:8A (Private)
Device type: general purpose
Running: Linux 2.4.X|2.5.X|2.6.X
OS details: Linux 2.4.18 - 2.6.7
```

```
Uptime 0.032 days (since Thu Mar 17 18:19:39 2005)

Nmap finished: 1 IP address (1 host up) scanned in 20.154 seconds
```

Figure 6: Nmap against the Host without IPTables

This result is very good, but after doing this, I realized that I had not started the IPTables. So that there would be more protection, let's take a look to a second run after IPtables:

```
C:\Program Files\nmap-3.81>nmap -sS –P0 –O 192.168.2.24
Starting nmap 3.81 ( http://www.insecure.org/nmap ) at 2005-03-17 20:21 SA Western Standard
Time
Interesting ports on hostname (192.168.2.24):
(The 1661 ports scanned but not shown below are in state: closed)
PORT    STATE SERVICE
22/tcp  open  ssh
MAC Address: 00:0E:9B:28:B6:8A (Private)
Device type: general purpose
Running: Linux 2.4.X|2.5.X|2.6.X
OS details: Linux 2.4.18 - 2.6.7
Uptime 0.032 days (since Thu Mar 17 18:19:39 2005)

Nmap finished: 1 IP address (1 host up) scanned in 23.794 seconds
```

Figure 7: Nmap against the Host with IPTables

The goal of this test was to identify any port in the Host open besides SSH (port 22). According to the last result, we passed this test.

Also, we wanted to test what happened when an attacker scans our Honeypot with Nmap. Remember the virtual machines we configured in the Honeypot (see Figure 2: Honeyd Configuration File). We have three virtual machines: Windows NT, Linux and Cisco router. Here are the Nmap results for each virtual machine:

```
C:\Program Files\nmap-3.81>nmap -sS –P0 –O 192.168.2.201

Starting nmap 3.81 ( http://www.insecure.org/nmap ) at 2005-03-17 21:04 SA Western Standard
Time
Interesting ports on 192.168.2.201:
(The 1659 ports scanned but not shown below are in state: closed)
PORT    STATE SERVICE
80/tcp  open  http
137/tcp open  netbios-ns
139/tcp open  netbios-ssn
143/tcp open  imap
```

```
MAC Address: 00:0E:9B:28:B6:8A (Private)
Device type: general purpose
Running: Microsoft Windows NT/2K/XP
OS details: Microsoft Windows 2000 Pro RC1 or Windows 2000 Advanced Server Beta3
, Microsoft Windows NT 4.0 Server SP5-SP6, Microsoft Windows NT 4.0 SP3
Uptime 628.489 days (since Sat Jun 28 09:31:25 2003)

Nmap finished: 1 IP address (1 host up) scanned in 684.804 seconds
```

Figure 8: Nmap against Windows NT Honeypot

```
C:\Program Files\nmap-3.81>nmap -sS –P0 –O 192.168.2.202

Starting nmap 3.81 ( http://www.insecure.org/nmap ) at 2005-03-17 20:32 SA Western Standard
Time
Interesting ports on 192.168.2.202:
(The 1660 ports scanned but not shown below are in state: closed)
PORT    STATE SERVICE
21/tcp  open  ftp
25/tcp  open  smtp
110/tcp open  pop3
MAC Address: 00:0E:9B:28:B6:8A (Private)
Device type: general purpose
Running: Linux 2.4.X
OS details: Linux 2.4.18 - 2.4.20 (x86)
Uptime 2.903 days (since Mon Mar 14 22:51:30 2005)

Nmap finished: 1 IP address (1 host up) scanned in 18.196 seconds
```

Figure 9: Nmap against Linux Honeypot

```
C:\Program Files\nmap-3.81>nmap -sS –P0 –O 192.168.2.203

Starting nmap 3.81 ( http://www.insecure.org/nmap ) at 2005-03-17 21:21 SA Western Standard
Time
Interesting ports on 192.168.2.203:
(The 1662 ports scanned but not shown below are in state: closed)
PORT   STATE SERVICE
23/tcp open  telnet
MAC Address: 00:0E:9B:28:B6:8A (Private)
Device type: router
Running: Cisco IOS 11.X|12.X
OS details: Cisco IOS 11.3 - 12.0(11)

Nmap finished: 1 IP address (1 host up) scanned in 15.082 seconds
```

Figure 10: Nmap against Cisco router Honeypot

Take a look at the services and the operating system detected by Nmap. They all do match with the services, configured for each Honeypot in the Honeyd

configuration file. This is very important because it is more difficult for the attacker to detect that those are fake machines by fingerprinting. That was the goal of this test and it passed.

4.2.3. Retina

Retina is a commercial software for Network Vulnerability Scans developed by eEye Digital Security (http://www.eeye.com). The Organization has a license and we are going to use it in order to run a network vulnerability assessment against the Host.

This tools runs in Windows platform. Installation of a running Retina is straightforward. This is a screen shot of how we configured a test against the Honeypots:

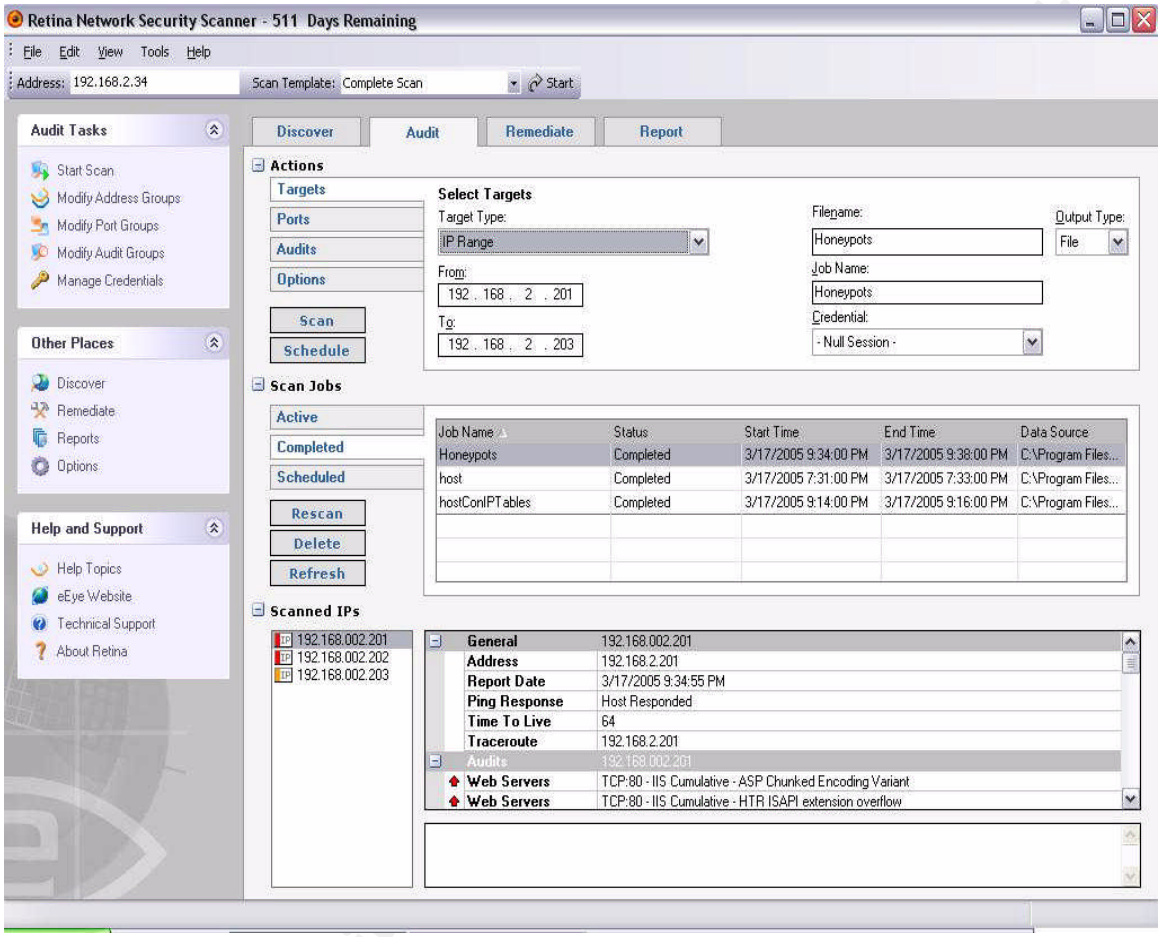

Figure 11: Retina test configuration screen

This screen corresponds to the Honeypots test. The results are in the Appendices (Retina Netwotk Scan Results / Honeypots). They are overwhelming because of the number of vulnerabilities listed in the report. The good news here is that the system detects the services configured in the virtual machines as if they were real. So of course, there are a lot of vulnerabilities associated with services as FTP, SMTP, Telnet, IMAP, and the other simulated services.

The goal here is similar to the Nmap execution against the Honeypots, to look if they seem to be real boxes. So for this test, the score was positive.

On the other hand, we ran a scan against the Host to look for vulnerabilities. As

we saw with Nmap, we did not find any vuonerabilities and the only port open is SSH. So this test has passed. The results are in the Appendices (Retina Netwotk Scan Results / The Host).

4.3. Password check

Although it is not supposed to be a lot of users created in a server with this kind of functionality, indeed this is our case, it is a good idea to check if there is some user with a weak password that an attacker could use to get access to the Host. Also, since the only service exposed by the Host is SSH, one of the first ways to get access is through password cracking.

For making this test we are going to use an open source software tool for password cracking called John the Ripper (http://www.openwall.com/john/). After downloading and unpacking it, the instructions for the installation are in the file ./src/INSTALL. It's just running a *make* command.

Also, there are other dictionaries that we could use in order to get more chance to catch a weak password. Here are a few ones that I downloaded from http://www.hackemate.com.ar/wordlists/: 31337.dic, cracklib.dic, female-names.zip, male-names.zip, spanish.dic, words-english-big.dic.

For this test we ran the tool with just one dictionary. The command and the results are bellow:

```
 [root@rs26s6 run]# ./john -wordfile:./dict/words-english-big.dic -rules /etc/shadow
Loaded 2 passwords with 2 different salts (FreeBSD MD5 [32/32])
guesses: 0  time: 0:00:15:55 14%  c/s: 2937  trying: 1amutter
guesses: 0  time: 0:00:25:18 20%  c/s: 2937  trying: pustulose!
guesses: 0  time: 0:00:26:49 21%  c/s: 2939  trying: fanmaker3
guesses: 0  time: 0:00:28:54 23%  c/s: 2940  trying: camisado7
guesses: 0  time: 0:00:32:45 25%  c/s: 2943  trying: pyic9
guesses: 0  time: 0:00:34:43 27%  c/s: 2943  trying: madotheca5
guesses: 0  time: 0:00:42:52 32%  c/s: 2936  trying: rabbleproof6
guesses: 0  time: 0:00:45:14 34%  c/s: 2935  trying: pint0
guesses: 0  time: 0:00:57:25 47%  c/s: 2931  trying: Cidanog
guesses: 0  time: 0:00:59:09 49%  c/s: 2931  trying: Eegamlrihw
```

```
guesses: 0  time: 0:01:01:14 52%  c/s: 2931  trying: predativE
guesses: 0  time: 0:02:04:41 100%  c/s: 2955  trying: Zyzzogetoning
```

Figure 12: John the ripper password cracking results

As you can see, the execution is very time consuming, it took about 2 hours. The results are positive, there were no weak passwords in our system.

4.4. Verify Honeyd Virtual Services Scripts

When we listed the vulnerabilities related with Honeypots, we mentioned that there could be risks in the scripts that simulated the services in the virtual machines. Since Honeyd is an open source software, there are people and groups that have developed and published their own scripts to simulate other services, vulnerabilities, etc. In this page (http://www.honeyd.org/contrib.php) you can see a good example of that. From a paranoid point of view, which in security is a good thing, some hacker could publish a fake Honeyd script that supposes to simulate some service. Maybe it seems to work well but behind the scenes is performing malicious activity. Sounds pretty much like a Back door to me. Or maybe it is supposed to be a simulated service, but it is actually the service that the attacker could use, (e.g. SMTP open relay server).

If we look inside these scripts, what they do is to print messages simulating responses from the service according to the input of the user connected.

The first thing I came out about verifying the scripts was connecting to the service and test as if we were in a real telnet session. This is a good test, but it is not effective because the attacker could simulate the answers you are waiting from that service. So it is more effective to open the source code of the scripts and see what is going on. You don't have to be a highly trained developer to figure out what is inside. Here is the source code of the SMTP service for Honeyd:

```
#!/bin/sh

#
# SMTP (Sendmail) Honeypot-Script intended for use with
# Honeyd from Niels Provos
# -> http://www.citi.umich.edu/u/provos/honeyd/
#
# Author: Maik Ellinger
```

```
# Last modified: 17/06/2002
# Version: 0.0.8
#
# Changelog:
# 0.0.7: - bugfix: sending correct CR/LF -> ToDO: correct all echo where necessary
#          - bugfix: handling of EHLO now clean
#          - bugfix: RCPT command response
#
# 0.0.4: - bugfix: suppress interpreting of control-characters
#

#set -x -v
DATE=`date`
host=`hostname`
domain=`dnsdomainname`
log=/var/log/honeyd/smtp-$1.log
MAILFROM="err"
EHELO="no"
RCPTTO="err"
echo "$DATE: SMTP started from $1 Port $2" >> $log
echo -e "220 $host.$domain ESMTP Sendmail 8.12.2/8.12.2/SuSE Linux 0.6; $DATE\r"
while read incmd parm1 parm2 parm3 parm4 parm5
do
        # remove control-characters
        incmd=`echo $incmd | sed s/[[:cntrl:]]//g`
        parm1=`echo $parm1 | sed s/[[:cntrl:]]//g`
        parm2=`echo $parm2 | sed s/[[:cntrl:]]//g`
        parm3=`echo $parm3 | sed s/[[:cntrl:]]//g`
        parm4=`echo $parm4 | sed s/[[:cntrl:]]//g`
        parm5=`echo $parm5 | sed s/[[:cntrl:]]//g`

        # convert to upper-case
        incmd_nocase=`echo $incmd | gawk '{print toupper($0);}'`
        #echo $incmd_nocase
        case $incmd_nocase in
          QUIT* )
                echo "220 2.0.0 $host.$domain closing connection"
          exit 0;;
          RSET* )
                echo "250 2.0.0 Reset state"
                ;;
          HELP* )
                echo "214-2.0.0 This is sendmail version 8,12,2"
                echo "214-2.0.0 Topics:"
                echo "214-2.0.0       HELO    EHLO    MAIL    RCPT    DATA"
                echo "214-2.0.0       RSET    NOOP    QUIT    HELP    VRFY"
                echo "214-2.0.0       EXPN    VERB    ETRN    DSN     AUTH"
                echo "214-2.0.0       STARTTLS"
```

```
                echo "214-2.0.0 For more info use \"HELP <topic>\"."
                echo "214-2.0.0 To report bugs in the implementation send email to"
                echo "214-2.0.0       sendmail-bugs@sendmail.org."
                echo "214-2.0.0 For local information send email to Postmaster at your site."
                echo "214 2.0.0 End of HELP info"
                ;;
        HELO* )
                if [ -n "$parm1" ]
                then
                  EHELO="ok"
                  echo "250 $host.$domain Hello $parm1[$1], pleased to meet you"
                else
                  echo "501 5.0.0 HELO requires domain address"
                fi
                ;;
        EHLO* )
                if [ -n "$parm1" ]
                then
                  EHELO="ok"
                  echo -e "250-$host.$domain Hello $parm1[$1], pleased to meet you\r"
                  echo -e "250-ENHANCEDSTATUSCODES\r"
          echo -e "250-PIPELINING\r"
                  echo -e "250-8BITMIME\r"
                  echo -e "250-SIZE\r"
                  echo -e "250-DSN\r"
                  echo -e "250-ETRN\r"
                  echo -e "250-DELIVERYBY\r"
                  echo -e "250 HELP\r"
                else
                  echo -e "501 5.0.0 EHLO requires domain address\r"
                fi
                ;;
        MAIL* )
                haveFROM=`echo $parm1 | gawk '{print toupper($0);}'`
                if [ "$haveFROM" == "FROM:" ]
                then
                                if [ -n "$parm2" ]
                                then
                                   MAILFROM="ok"
                                   echo "250 2.1.0 $parm2 $parm3 $parm4... Sender ok"
                                else
                                      echo "501 5.5.2 Syntax error in parameters scanning
\"$parm2\""
                                   MAILFROM="err"
                                fi
                else
                  echo "501 5.5.2 Syntax error in parameters scanning \"\""
                fi
```

```
                ;;
        RCPT* )
                #echo $MAILFROM
                if [ "$MAILFROM" == "ok"  ]
                then
                        haveTO=`echo $parm1 | gawk '{print toupper($0);}'`
                        if [ "$haveTO" == "TO:"  ]
                        then
                                if [ -n "$parm2" ]
                                then
                                   RCPTTO="ok"
                                     echo "553 sorry, that domain isn't in my list of allowed
rcpthosts (#6.7.1)"
                                else
                                   echo "501 5.5.2 Syntax error in parameters scanning \"\""
                                   RCPTTO="err"
                                fi
                        fi
                else
                  echo "503 5.0.0 Need MAIL before RCPT"
                fi
                ;;
        STARTTLS* )
                echo "454 4.3.3 TLS not available after start"
                ;;
        NOOP* )
                echo "250 2.0.0 OK"
                ;;
        AUTH* )
                echo "503 AUTH mechanism not available"
                ;;
        * )
                echo "500 5.5.1 Command unrecognized: \"$incmd\""
                ;;
    esac
    echo "$incmd $parm1 $parm2 $parm3 $parm4 $parm5" >> $log
done
```

Figure 13: Script of SMTP simulated service for Honeyd

As we can see, there are only if-sentences to see what the input of the user (attacker) was and then print an echo message with the response to that input as if it were the real service.

If we open a script that supposes to simulate a service and see functions calls as socket, write, read, etc., we should get rid of it.

This check must be done in every single script. We can say that the test passed with the SMTP script.

4.5. Logging

The last test I recommend is to check if all the activity we did in previous tests against the Honeypot and Host was logged. Is the logged information useful and enough to identify the source and have and idea of what was done?

First, we are going to examine the Honeyd log generated. The following is a fragment of the log with data captured during the Nmap sessions.

```
2005-03-17-20:25:02.0433 honeyd packet log started ------
2005-03-17-20:25:25.0530 icmp(1) - 192.168.2.34 192.168.2.201: 8(0): 28
2005-03-17-20:25:25.0530 tcp(6) - 192.168.2.34 59910 192.168.2.201 80: 40 A
2005-03-17-20:25:25.0532 icmp(1) - 192.168.2.34 192.168.2.201: 8(0): 28
2005-03-17-20:25:25.0534 tcp(6) - 192.168.2.34 59911 192.168.2.201 80: 40 A
2005-03-17-20:25:33.0000 tcp(6) - 192.168.2.34 59889 192.168.2.201 3389: 40 S
2005-03-17-20:25:33.0000 tcp(6) - 192.168.2.34 59889 192.168.2.201 443: 40 S
2005-03-17-20:25:33.0003 tcp(6) - 192.168.2.34 59889 192.168.2.201 1723: 40 S
2005-03-17-20:25:33.0004 tcp(6) - 192.168.2.34 59889 192.168.2.201 22: 40 S
2005-03-17-20:25:33.0006 tcp(6) S 192.168.2.34 59889 192.168.2.201 80
2005-03-17-20:25:33.0008 tcp(6) - 192.168.2.34 59889 192.168.2.201 25: 40 S
2005-03-17-20:25:33.0010 tcp(6) - 192.168.2.34 59889 192.168.2.201 636: 40 S
2005-03-17-20:25:33.0012 tcp(6) - 192.168.2.34 59889 192.168.2.201 53: 40 S
2005-03-17-20:25:33.0014 tcp(6) - 192.168.2.34 59889 192.168.2.201 256: 40 S
2005-03-17-20:25:33.0015 tcp(6) - 192.168.2.34 59889 192.168.2.201 554: 40 S
2005-03-17-20:25:33.0018 tcp(6) - 192.168.2.34 59889 192.168.2.201 23: 40 S
2005-03-17-20:25:33.0020 tcp(6) - 192.168.2.34 59889 192.168.2.201 113: 40 S
2005-03-17-20:25:33.0021 tcp(6) - 192.168.2.34 59889 192.168.2.201 389: 40 S
2005-03-17-20:25:33.0022 tcp(6) - 192.168.2.34 59889 192.168.2.201 21: 40 S
2005-03-17-20:25:33.0026 tcp(6) - 192.168.2.34 59889 192.168.2.201 2201: 40 S
2005-03-17-20:25:33.0026 tcp(6) - 192.168.2.34 59889 192.168.2.201 57: 40 S
2005-03-17-20:25:33.0027 tcp(6) - 192.168.2.34 59889 192.168.2.201 779: 40 S
2005-03-17-20:25:33.0029 tcp(6) - 192.168.2.34 59889 192.168.2.201 964: 40 S
```

Figure 14: Honeyd logging Nmap activity

These logs don't offer too much information about activity, just the connection. This is a good log to trigger alarms when an entry is added, remember that any connection against the Honeypot is considered suspicious.

The Snort Alerts are actually potential malicious activity against the Host. They show information about what the event was, and links to other resources to make a deep investigation.

```
 [**] [1:1228:7] SCAN nmap XMAS [**]
[Classification: Attempted Information Leak] [Priority: 2]
03/17-20:32:21.420484 0:3:47:9D:1E:7A -> 0:E:9B:28:B6:8A type:0x800 len:0x4A
192.168.2.34:48979 -> 192.168.2.202:1 TCP TTL:51 TOS:0x0 ID:35345 IpLen:20 DgmLen:60
**U*P**F Seq: 0x72F00BD5  Ack: 0x0  Win: 0x1000  TcpLen: 40  UrgPtr: 0x0
TCP Options (5) => WS: 10 NOP MSS: 265 TS: 1061109567 0 EOL
[Xref => http://www.whitehats.com/info/IDS30]

[**] [122:1:0] (portscan) TCP Portscan [**]
03/17-21:04:02.679203 4D:41:43:44:41:44 -> 4D:41:43:44:41:44 type:0x800
len:0xAE192.168.2.34 -> 192.168.2.201 PROTO255 TTL:0 TOS:0x0 ID:1738 IpLen:20
DgmLen:160
[**] [1:1421:11] SNMP AgentX/tcp request [**]
[Classification: Attempted Information Leak] [Priority: 2]
03/17-21:04:15.698691 0:3:47:9D:1E:7A -> 0:E:9B:28:B6:8A type:0x800 len:0x3C
192.168.2.34:48127 -> 192.168.2.201:705 TCP TTL:49 TOS:0x0 ID:33345 IpLen:20
DgmLen:40
```

Figure 15: Snort alerting Nmap activity

Snort has another nice feature which is generating separate files for logging, just the sessions in ASCII format. We can locate these files in /var/logs/snort/SESSION*. Here is a sample IMAP sessions against the Honeypot:

```
* OK Microsoft Exchange 2000 IMAP4rev1 server version 6.0.6249.0 (bps-pc9.) ready.
1 login hcira pass4imap
1 NO Logon failure: unknown user name or bad password.
2 logout
* BYE Microsoft Exchange 2000 IMAP4rev1 server version 6.0.6249.0 signing off.
2 OK LOGOUT completed.
```

Figure 16: Snort logging IMAP session

Snort also has a log file with all the traffic inbound and outbound of the Host. Here are two fragments of the same IMAP session above:

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

[**] Snort Unmatched [**]
03/15-21:25:59.463828 0:E:9B:28:B6:8A -> 0:3:47:9D:1E:7A type:0x800 len:0x89
192.168.2.201:143 -> 192.168.2.34:1060 TCP TTL:64 TOS:0x0 ID:70 IpLen:20 DgmLen:123
```

```
***A**** Seq: 0x5236941D  Ack: 0xC4BCFCE9  Win: 0x2017  TcpLen: 20
2A 20 4F 4B 20 4D 69 63 72 6F 73 6F 66 74 20 45  * OK Microsoft E
78 63 68 61 6E 67 65 20 32 30 30 30 20 49 4D 41  xchange 2000 IMA
50 34 72 65 76 31 20 73 65 72 76 65 72 20 76 65  P4rev1 server ve
72 73 69 6F 6E 20 36 2E 30 2E 36 32 34 39 2E 30  rsion 6.0.6249.0
20 28 62 70 73 2D 70 63 39 2E 29 20 72 65 61 64   (bps-pc9.) read
79 2E 0A                                         y..

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

.
.

[**] Snort Unmatched [**]
03/15-21:26:12.409257 0:E:9B:28:B6:8A -> 0:3:47:9D:1E:7A type:0x800 len:0x6D
192.168.2.201:143 -> 192.168.2.34:1060 TCP TTL:64 TOS:0x0 ID:100 IpLen:20 DgmLen:95
***A**** Seq: 0x52369470  Ack: 0xC4BCFD07  Win: 0x2017  TcpLen: 20
31 20 4E 4F 20 4C 6F 67 6F 6E 20 66 61 69 6C 75  1 NO Logon failu
72 65 3A 20 75 6E 6B 6E 6F 77 6E 20 75 73 65 72  re: unknown user
20 6E 61 6D 65 20 6F 72 20 62 61 64 20 70 61 73   name or bad pas
73 77 6F 72 64 2E 0A                             sword..

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
```

Figure 17: Snort capturing IMAP session packets

There is an easier way to query and look at the whole session stream using Ethereal (http://www.ethereal.com/). This is a very popular network protocol analyzer that we are going to use to look at the log file generated by TCPDump. When we ran TCPDump before to start capturing traffic, we used the flag –w in order to generate the log in a format that tools like Ethereal can read.

In order to look at the log generated by TCPDump I had to copy the file from `/var/logs/tcpdump/tcpdump`*.log* to a Windows machine with Ethereal installed on it. Then start Ethereal and open the file `tcpdump.log`. This is the screen with the log loaded in Ethereal:

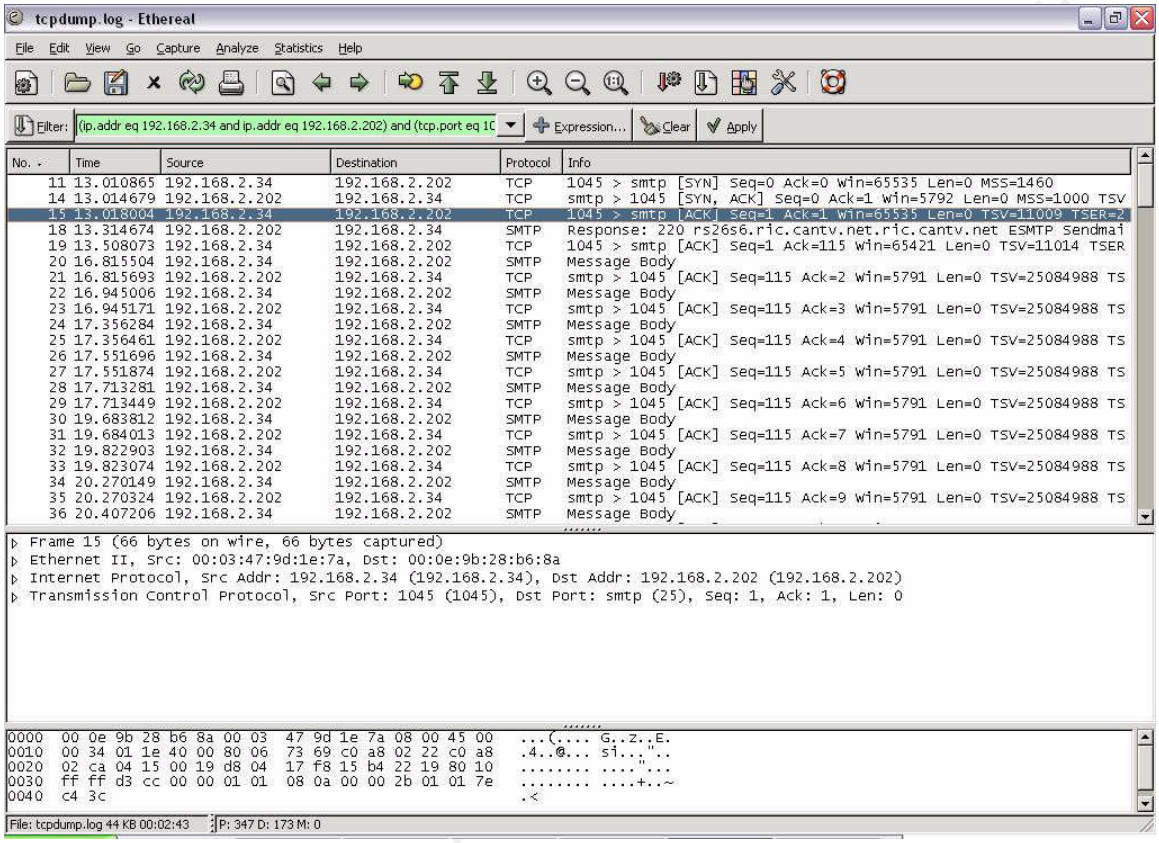

Figure 18. Ethereal session with TCPDump.log opened

The line highlighted belongs to a smtp session opened against one of the Honeypots. If we go to Analyze menu and the Follow TCP Stream, we are going to see the whole SMTP session as follows:

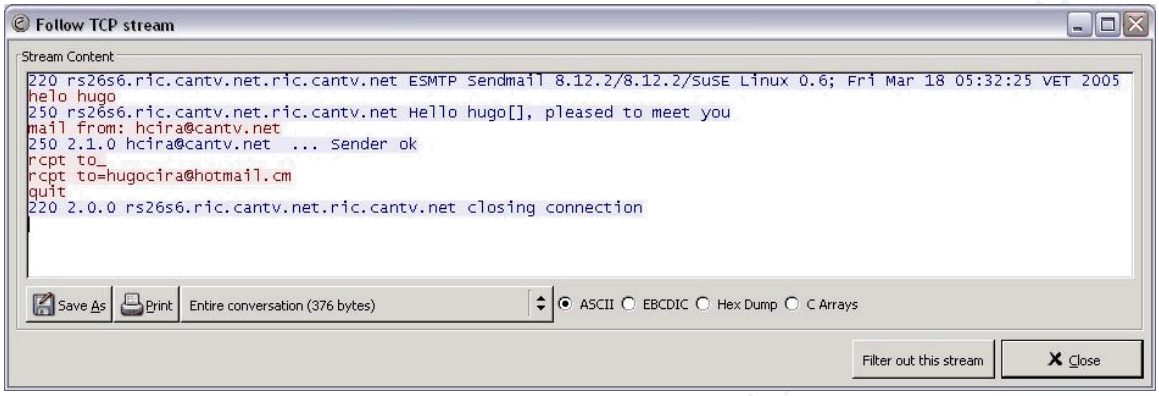

Figure 19: SMTP session captured with TCPDump "Follow TCP Stream" feature

The goal of this test was to verify that all the activity going on against the Host and Honeypot is actually logged. If attackers gains control over the Honeypot box and we never realize it, we are going to be in big trouble. This test passed.

5. Conclusions

One of the goals of this analysis was detecting vulnerabilities that may allow attackers to gain access to the system and then compromise another networks. The only access to the system must be done through the virtual machines exposed by Honeyd. With the checklist we ran inside the host there was evidence that there is work to do by the administrators in order to make that system more secured and mitigate the risks associated.

However, the big picture from outside the host (corporate network) looks very well; we scan the server with two different scanners (Nmap and Retina) and the only thing we found opened is SSH. This is good news for us because we want all connections going against the Honeypot; this way the organization is able to capture evidence of corporate desktops with virus, internal attackers, external attackers, and anything that jeopardize the organization's information assets.

The other goal was testing the integrity of the system, checking on data control and data capture mechanisms, and the content of the scripts used to simulate the services in the Honeypots. For the system analyzed, these tests were 100% passed. However, the recommendation in the case of data capture and data control mechanisms is that they must be out of the Honeypot server for more security. If an attacker takes control of the system with Administrator privileges, he could kill the process of logging and IPTables in order to be free to do attacks to internal and external networks. This separation of functionalities in layers is named by the Honey Net Project organization as GenII Honeynets.

6. References

SANS Institute (2004). *Track 8 – Systems Forensics, Investigation and Response*. Book 8.5.

Spitzner, L. (2003). *Honeypots, Tracking Hackers.* Boston: Addison Wesley. http://www.tracking-hackers.com/

Spitzner, L. (January 20, 2003). *Open Source Honeypots: Learning with Honeyd*. SecurityFocus. http://www.securityfocus.com/printable/infocus/1659.

Spitzner, L. (March 12, 2003). *Open Source Honeypots, Part Two: Deploying Honeyd in the Wild*. SecurityFocus. http://www.securityfocus.com/printable/infocus/1675.

The Honey Net Project (2004). *Know Your Enemy. Learning About Security Threats (*Second edition). Boston: Addison Wesley.

7. Appendices

7.1. CIS Benchmark Scoring Tool Results

7.1.1. Final Score

```
Ending run at time: Thu Mar 17 19:16:19 2005

       Final rating = 6.00 / 10.00
```

7.1.2. CIS Benchmark Scoring Tool Output

```
[root@rs26s6 CIS]# ./cis-scan
*******************************************************************************
******************** CIS Security Benchmark Checker v1.4.2 ********************
*                                                                             *
* Lead Developer: Jay Beale         *
* Benchmark Coordinator and Gadfly: Hal Pomeranz      *
*                                                                             *
* Copright 2001 - 2003 The Center for Internet Security  www.cisecurity.org *
*                                                                             *
* Please send feedback to linux-scan@cisecurity.org.                    *
*******************************************************************************

        Investigating system...this will take a few minutes...

                              ******

Now a final check for non-standard world-writable files, Set-UID and Set-GID
programs -- this can take a whole lot of time if you have a large filesystem.
Your score if there are no extra world-writable files or SUID/SGID programs
found will be 6.31 / 10.00 .  If there are extra SUID/SGID programs or
world-writable files, your score could be as low as 6.00 / 10.00 .

       You can hit CTRL-C at any time to stop at this remaining step.

The preliminary log can be found at: ./cis-most-recent-log

                              ******


          Rating = 6.00 / 10.00
```

```
*******************************************************************************
To learn more about the results, do the following:

  All results/diagnostics:
      more ./cis-ruler-log.20050317-19:12:30.6523
  Positive Results Only:
      egrep "^Positive" ./cis-ruler-log.20050317-19:12:30.6523
  Negative Results Only:
      egrep "^Negative" ./cis-ruler-log.20050317-19:12:30.6523

For each item that you score or fail to score on, please reference the
corresponding item in the CIS Benchmark Document.


  For additional instructions/support, please reference the CIS web page:

                  http://www.cisecurity.org
```

7.1.3. Positives

```
Positive: 1.1 System appears to have been patched within the last month.
Positive: 1.2 System is running sshd and it's configured well.
Positive: 2.1 inetd/xinetd is not listening on any of the miscellaneous ports checked in this item.
Positive: 2.2 telnet is deactivated.
Positive: 2.3 ftp is deactivated.
Positive: 2.4 rsh, rcp and rlogin are deactivated.
Positive: 2.5 tftp is deactivated.
Positive: 2.6 imap is deactivated.
Positive: 2.7 POP server is deactivated.
Positive: 3.1 Found a good daemon umask of 022 in /etc/rc.d/init.d/functions.
Positive: 3.7 Windows compatibility servers (samba) have been deactivated.
Positive: 3.8 NFS Server script nfs is deactivated.
Positive: 3.10 NIS Client processes are deactivated.
Positive: 3.11 NIS Server processes are deactivated.
Positive: 3.15 Web server is deactivated.
Positive: 3.16 SNMP daemon is deactivated.
Positive: 3.17 DNS server is deactivated.
Positive: 3.18 SQL database server is deactivated.
Positive: 3.19 Webmin GUI-based system administration daemon deactivated.
Positive: 3.20 Squid web cache daemon deactivated.
Positive: 5.1 syslog captures authpriv messages.
Positive: 5.2 FTP server is configured to do full logging.
Positive: 5.3 All logfile permissions and owners match benchmark recommendations.
Positive: 6.1 All appropriate partitions are mounted nodev.
Positive: 6.2 /etc/fstab mounts all removable filesystems nosuid and nodev.
Positive: 6.4 password and group files have right permissions and owners.
```

Positive: 6.5 all temporary directories have sticky bits set.
Positive: 7.1 rhosts authentication totally deactivated in PAM.
Positive: 7.2 /etc/hosts.equiv and root's .rhosts/.shosts files either don't exist, are zero size or are links to /dev/null.
Positive: 7.3 FTP daemons do not permit system users to use FTP.
Positive: 7.4 X11 Server is blocked from listening on TCP port 6000.
Positive: 7.12 NFS server restricts clients to privileged ports.
Positive: 8.2 All users have passwords
Positive: 8.4 There were no +: entries in passwd, shadow or group maps.
Positive: 8.5 Only one UID 0 account AND it is named root.
Positive: 8.6 root's PATH is clean of group/world writable directories or the current-directory link.
Positive: 8.7 No user's home directory is world or group writable.
Positive: 8.8 No group or world-writable dotfiles in user home directories!
Positive: 8.9 No user has a .netrc file.

7.1.4. Negatives

Negative: 3.2 xinetd is still active.
Negative: 3.3 Mail daemon is still listening on TCP 25.
Negative: 3.4 Graphical login not deactivated.
Negative: 3.5 X Font Server (xfs) script has not been deactivated
Negative: 3.6 apmd not deactivated.
Negative: 3.6 gpm not deactivated.
Negative: 3.6 isdn not deactivated.
Negative: 3.9 NFS script nfslock not deactivated.
Negative: 3.9 NFS script autofs not deactivated.
Negative: 3.12 RPC rc-script (portmap) has not been deactivated.
Negative: 3.13 netfs rc script not deactivated.
Negative: 3.14 cups (printing daemon) not deactivated.
Negative: 3.21 Kudzu hardware detection program has not been deactivated.
Negative: 4.1 /proc/sys/net/ipv4/conf/eth0/secure_redirects should be set to 0.
Negative: 4.1 /proc/sys/net/ipv4/conf/lo/secure_redirects should be set to 0.
Negative: 4.1 /proc/sys/net/ipv4/conf/eth0/accept_redirects should be set to 0.
Negative: 4.1 /proc/sys/net/ipv4/conf/lo/accept_redirects should be set to 0.
Negative: 4.1 /proc/sys/net/ipv4/tcp_max_syn_backlog should be at least 4096 to handle SYN floods.
Negative: 4.2 /proc/sys/net/ipv4/conf/eth0/send_redirects should be set to 0.
Negative: 4.2 /proc/sys/net/ipv4/conf/lo/send_redirects should be set to 0.
Negative: 6.3 PAM allows users to mount removable media: <floppy>. (/etc/security/console.perms)
Negative: 6.3 PAM allows users to mount removable media: <cdrom>. (/etc/security/console.perms)
Negative: 6.3 PAM allows users to mount removable media: <pilot>. (/etc/security/console.perms)
Negative: 6.3 PAM allows users to mount removable media: <jaz>. (/etc/security/console.perms)
Negative: 6.3 PAM allows users to mount removable media: <zip>. (/etc/security/console.perms)
Negative: 6.3 PAM allows users to mount removable media: <ls120>. (/etc/security/console.perms)

Negative: 6.3 PAM allows users to mount removable media: <camera>. (/etc/security/console.perms)
Negative: 6.3 PAM allows users to mount removable media: <memstick>. (/etc/security/console.perms)
Negative: 6.3 PAM allows users to mount removable media: <flash>. (/etc/security/console.perms)
Negative: 6.3 PAM allows users to mount removable media: <diskonkey>. (/etc/security/console.perms)
Negative: 6.3 PAM allows users to mount removable media: <rem_ide>. (/etc/security/console.perms)
Negative: 6.3 PAM allows users to mount removable media: <rio500>. (/etc/security/console.perms)
Negative: 6.3 PAM allows users to mount removable media: <pmu>. (/etc/security/console.perms)
Negative: 6.3 PAM allows users to mount removable media: <bluetooth>. (/etc/security/console.perms)
Negative: 6.3 PAM allows users to mount removable media: <raw1394>. (/etc/security/console.perms)
Negative: 7.5 Couldn't open cron.allow
Negative: 7.5 Couldn't open at.allow
Negative: 7.6 The permissions on /etc/crontab are not sufficiently restrictive.
Negative: 7.7 No Authorized Only message in /etc/motd.
Negative: 7.7 No Authorized Only message in Gnome's /etc/X11/gdm/gdm.conf.
Negative: 7.7 No Authorized Only banner for krb5-telnet in file /etc/xinetd.d/krb5-telnet.
Negative: 7.7 No Authorized Only banner for klogin in file /etc/xinetd.d/klogin.
Negative: 7.7 No Authorized Only banner for eklogin in file /etc/xinetd.d/eklogin.
Negative: 7.7 No Authorized Only banner for gssftp in file /etc/xinetd.d/gssftp.
Negative: 7.7 No Authorized Only banner for kshell in file /etc/xinetd.d/kshell.
Negative: 7.8 xinetd either requires global 'only-from' statement or one for each service.
Negative: 7.9 /etc/securetty has a non console or tty 1-6 line: vc/7.
Negative: 7.9 /etc/securetty has a non console or tty 1-6 line: vc/8.
Negative: 7.9 /etc/securetty has a non console or tty 1-6 line: vc/9.
Negative: 7.9 /etc/securetty has a non console or tty 1-6 line: vc/10.
Negative: 7.9 /etc/securetty has a non console or tty 1-6 line: vc/11.
Negative: 7.9 /etc/securetty has a non console or tty 1-6 line: tty7.
Negative: 7.9 /etc/securetty has a non console or tty 1-6 line: tty8.
Negative: 7.9 /etc/securetty has a non console or tty 1-6 line: tty9.
Negative: 7.9 /etc/securetty has a non console or tty 1-6 line: tty10.
Negative: 7.9 /etc/securetty has a non console or tty 1-6 line: tty11.
Negative: 7.10 GRUB isn't password-protected.
Negative: 7.11 /etc/inittab needs a /sbin/sulogin line for single user mode.
Negative: 8.1 bin has a valid shell of /sbin/nologin. Remember, the /sbin/nologin shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 daemon has a valid shell of /sbin/nologin. Remember, the /sbin/nologin shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 adm has a valid shell of /sbin/nologin. Remember, the /sbin/nologin shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 lp has a valid shell of /sbin/nologin. Remember, the /sbin/nologin shell, when found

in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 mail has a valid shell of /sbin/nologin. Remember, the /sbin/nologin shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 news has a valid shell of /bin/sh. Remember, an empty shell field in /etc/passwd signifies /bin/sh.
Negative: 8.1 uucp has a valid shell of /sbin/nologin. Remember, the /sbin/nologin shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 operator has a valid shell of /sbin/nologin. Remember, the /sbin/nologin shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 games has a valid shell of /sbin/nologin. Remember, the /sbin/nologin shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 gopher has a valid shell of /sbin/nologin. Remember, the /sbin/nologin shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 ftp has a valid shell of /sbin/nologin. Remember, the /sbin/nologin shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 nobody has a valid shell of /sbin/nologin. Remember, the /sbin/nologin shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 dbus has a valid shell of /sbin/nologin. Remember, the /sbin/nologin shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 vcsa has a valid shell of /sbin/nologin. Remember, the /sbin/nologin shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 nscd has a valid shell of /sbin/nologin. Remember, the /sbin/nologin shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 rpm has a valid shell of /sbin/nologin. Remember, the /sbin/nologin shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 haldaemon has a valid shell of /sbin/nologin. Remember, the /sbin/nologin shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 netdump has a valid shell of /bin/bash.
Negative: 8.1 sshd has a valid shell of /sbin/nologin. Remember, the /sbin/nologin shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 rpc has a valid shell of /sbin/nologin. Remember, the /sbin/nologin shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 rpcuser has a valid shell of /sbin/nologin. Remember, the /sbin/nologin shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 mailnull has a valid shell of /sbin/nologin. Remember, the /sbin/nologin shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 smmsp has a valid shell of /sbin/nologin. Remember, the /sbin/nologin shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 pcap has a valid shell of /sbin/nologin. Remember, the /sbin/nologin shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 xfs has a valid shell of /sbin/nologin. Remember, the /sbin/nologin shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 ntp has a valid shell of /sbin/nologin. Remember, the /sbin/nologin shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 gdm has a valid shell of /sbin/nologin. Remember, the /sbin/nologin shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.3 User hcira should have a minimum password life of at least 7 days.
Negative: 8.3 User hcira should have a maximum password life of between 1 and 90 days.

```
Negative: 8.3 /etc/login.defs value PASS_MAX_DAYS = 99999, but should not exceed 90.
Negative: 8.3 /etc/login.defs value PASS_MIN_DAYS = 0, but should not be less than 7.
Negative: 8.3 /etc/login.defs value PASS_MIN_LEN = 5, but should be at least 6.
Negative: 8.10 Current umask setting in file /etc/profile is 000 -- it should be stronger to block world-read/write/execute.
Negative: 8.10 Current umask setting in file /etc/profile is 000 -- it should be stronger to block group-read/write/execute.
Negative: 8.10 Current umask setting in file /etc/csh.login is 000 -- it should be stronger to block world-read/write/execute.
Negative: 8.10 Current umask setting in file /etc/csh.login is 000 -- it should be stronger to block group-read/write/execute.
Negative: 8.10 Current umask setting in file /etc/bashrc is 022 -- it should be stronger to block world-read/write/execute.
Negative: 8.10 Current umask setting in file /etc/bashrc is 022 -- it should be stronger to block group-read/write/execute.
Negative: 8.10 Current umask setting in file /etc/csh.cshrc is 002 -- it should be stronger to block world-read/write/execute.
Negative: 8.10 Current umask setting in file /etc/csh.cshrc is 002 -- it should be stronger to block group-read/write/execute.
Negative: 8.10 Current umask setting in file /root/.bash_profile is 000 -- it should be stronger to block world-read/write/execute.
Negative: 8.10 Current umask setting in file /root/.bash_profile is 000 -- it should be stronger to block group-read/write/execute.
Negative: 8.10 Current umask setting in file /root/.bashrc is 000 -- it should be stronger to block world-read/write/execute.
Negative: 8.10 Current umask setting in file /root/.bashrc is 000 -- it should be stronger to block group-read/write/execute.
Negative: 8.10 Current umask setting in file /root/.cshrc is 000 -- it should be stronger to block world-read/write/execute.
Negative: 8.10 Current umask setting in file /root/.cshrc is 000 -- it should be stronger to block group-read/write/execute.
Negative: 8.10 Current umask setting in file /root/.tcshrc is 000 -- it should be stronger to block world-read/write/execute.
Negative: 8.10 Current umask setting in file /root/.tcshrc is 000 -- it should be stronger to block group-read/write/execute.
Negative: 8.11 Coredumps aren't deactivated.
```

7.2. Retina Netwotk Scan Results

7.2.1. The Host

Retina - Network Security Scanner

Network Vulnerability Assessment & Remediation Management

Saturday, March 19, 2005

NETWORK ANALYSIS RESULTS
Report Summary

Scanner Name
Retina

Machines Scanned
1

Scanner Version
5.2.6.1208

Vulnerabilities Total
0

Scan Start Date
3/19/2005

High Risk Vulnerabilities
0

Scan Start Time
6:54:00 PM

Medium Risk Vulnerabilities
0

Scan Duration
0h 2m 0s

Low Risk Vulnerabilities
0

TOP 20 OPEN PORTS

The following is an overview of the top 20 open ports on your network.

Rank
Port Number
Description
Count

1.
TCP:22
SSH - SSH (Secure Shell) Remote Login Protocol
1

7.2.2. Honeypot

These are the results of the network scans over the three Honeypots.

Retina - Network Security Scanner
Network Vulnerability Assessment & Remediation Management

Thursday, March 17, 2005

Anonymous Write

Risk Level:

High

Category:

FTP Servers

Description:

Giving an anonymous user the ability to write to your disk is not recommended as it can lead to the compromise of your system.

How To Fix:

Follow your FTP server instructions on how to disable anonymous write access.

CVE:

CAN-1999-0527 (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0527)
 - The permissions for system-critical data in an anonymous FTP account are inappropriate. For example, the root directory is writeable by world, a real password file is obtainable, or executable commands such as "ls" can be overwritten.
CAN-1999-0497 (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0497)
 - Anonymous FTP is enabled

IAV:

Affected Machines:

IP Address

Machine Name

OS

☐

192.168.002.202

unknown

Linux 2.4.18 - 2.4.20 (X86)

Retina - Network Security Scanner
Network Vulnerability Assessment & Remediation Management

Thursday, March 17, 2005

IIS Cumulative - ASP Chunked Encoding Variant

Risk Level:

High

Category:

Web Servers

Description:

There exists a variant buffer overflow vulnerability within how Microsoft IIS handles chunked encoding requests.

How To Fix:

Install the Microsoft patch.

Related Links:

Microsoft Knowledge Base Article Q319733 (http://support.microsoft.com/default.aspx?scid=kb;en-us;Q319733)
Microsoft Security Bulletin MS02-018 (http://www.microsoft.com/technet/security/bulletin/MS02-018.mspx)

CVE:

CVE-2002-0147 (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2002-0147)
- Buffer overflow in the ASP data transfer mechanism in Internet Information Server (IIS) 4.0, 5.0, and 5.1 allows remote attackers to cause a denial of service or execute code, aka "Microsoft-discovered variant of Chunked Encoding buffer overrun."

IAV:

BugtraqID:

4490 (http://www.securityfocus.com/bid/4490)
- Microsoft IIS Chunked Encoding Heap Overflow Variant Vulnerability

Affected Machines:

IP Address

Machine Name

OS

☐
192.168.002.201
unknown
Microsoft Windows NT 4.0 SP3, Microsoft Windows NT 4.0 Server SP5-SP6, Microsoft Windows 2000 Professional RC1 or Windows 2000 Advanced Server Beta3

Retina - Network Security Scanner
Network Vulnerability Assessment & Remediation Management

Thursday, March 17, 2005

IIS Cumulative - HTR ISAPI extension overflow

Risk Level:

High

Category:

Web Servers

Description:

There exists a buffer overflow vulnerability within the Microsoft IIS .htr ISAPI filter. Attackers can potentially leverage this vulnerability to execute malicious code remotely on your web server.

How To Fix:

Install the appropriate Microsoft patch.

Related Links:

Microsoft Knowledge Base Article Q319733 (http://support.microsoft.com/default.aspx?scid=kb;en-us;Q319733)
Microsoft Security Bulletin MS02-018 (http://www.microsoft.com/technet/security/bulletin/MS02-018.mspx)

CVE:

CVE-2002-0071 (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2002-0071)
- Buffer overflow in the ism.dll ISAPI extension that implements HTR scripting in Internet Information Server (IIS) 4.0 and 5.0 allows attackers to cause a denial of service or execute arbitrary code via HTR requests with long variable names.

IAV:

BugtraqID:

4474 (http://www.securityfocus.com/bid/4474)
- Microsoft IIS HTR ISAPI Extension Buffer Overflow Vulnerability

Affected Machines:

IP Address

Machine Name

OS

☐

192.168.002.201

unknown

Microsoft Windows NT 4.0 SP3, Microsoft Windows NT 4.0 Server SP5-SP6, Microsoft Windows 2000 Professional RC1 or Windows 2000 Advanced Server Beta3

Retina - Network Security Scanner
Network Vulnerability Assessment & Remediation Management

Thursday, March 17, 2005

IIS Cumulative - HTTP Header Overflow

Risk Level:

High

Category:

Web Servers

Description:

There exists a buffer overflow within how Microsoft IIS handles HTTP header data. Attackers can exploit this vulnerability in order to remotely execute code on a susceptible web server.

How To Fix:

Install the appropriate Microsoft patch.

Related Links:

Microsoft Knowledge Base Article Q319733 (http://support.microsoft.com/default.aspx?scid=kb;en-us;Q319733)
Microsoft Security Bulletin MS02-018 (http://www.microsoft.com/technet/security/bulletin/MS02-018.mspx)

CVE:

CVE-2002-0150 (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2002-0150)
- Buffer overflow in Internet Information Server (IIS) 4.0, 5.0, and 5.1 allows remote attackers to spoof the safety check for HTTP headers and cause a denial of service or execute arbitrary code via HTTP header field values.

IAV:

BugtraqID:

4476 (http://www.securityfocus.com/bid/4476)
- Microsoft IIS HTTP Header Field Delimiter Buffer Overflow Vulnerability

Affected Machines:

IP Address

Machine Name

OS

☐
192.168.002.201
unknown
Microsoft Windows NT 4.0 SP3, Microsoft Windows NT 4.0 Server SP5-SP6, Microsoft Windows 2000 Professional RC1 or Windows 2000 Advanced Server Beta3

Retina - Network Security Scanner
Network Vulnerability Assessment & Remediation Management

Thursday, March 17, 2005

MDAC RDS is accessible

Risk Level:

High

Category:

Web Servers

Description:

Microsoft Data Access Components (MDAC) Remote Data Services is enabled and accessible remotely. Various vulnerabilities have been discovered in RDS in the past that permit remote attackers to compromise the vulnerable web server.

How To Fix:

We recommend unmapping RDS from the Internet Services Manager if you are not using its functionality. Otherwise, verify that you have the most recent version of MDAC and all appropriate hotfixes installed.

Related Links:

Microsoft Security Bulletin MS99-025 (http://www.microsoft.com/technet/security/bulletin/MS99-025.mspx)
Microsoft Security Bulletin MS02-065 (http://www.microsoft.com/technet/security/bulletin/MS02-065.mspx)

CVE:

CVE-1999-1011 (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-1011)
- The Remote Data Service (RDS) DataFactory component of Microsoft Data Access Components (MDAC) in IIS 3.x and 4.x exposes unsafe methods, which allows remote attackers to execute arbitrary commands.
CVE-2002-1142 (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2002-1142)
- When msadc.dll is accessed over HTTP. This exploit was known as the RDS exploit.

IAV:

BugtraqID:

6214 (http://www.securityfocus.com/bid/6214)
- Microsoft Data Access Components RDS Buffer Overflow Vulnerability

Affected Machines:

IP Address
Machine Name
OS

☐
192.168.002.201
unknown

Retina - Network Security Scanner
Network Vulnerability Assessment & Remediation Management

Thursday, March 17, 2005

NT IIS MDAC RDS Vulnerability

Risk Level:

High

Category:

CGI Scripts

Description:

The Remote Data Service (RDS) DataFactory component of Microsoft Data Access Components (MDAC) in IIS 3.0 and 4.0 exposes unsafe methods, which can be exploited by remote attackers to execute arbitrary commands with SYSTEM level privileges.

How To Fix:

Remove the /msadc directory and IIS virtual mapping. Also install MDAC 2.1 SP2 or higher. If you have this patch installed and are seeing this vulnerability then please disregard.

Related Links:

MDAC Download Page (http://www.microsoft.com/data/download.htm)

CVE:

CVE-1999-1011 (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-1011)
- The Remote Data Service (RDS) DataFactory component of Microsoft Data Access Components (MDAC) in IIS 3.x and 4.x exposes unsafe methods, which allows remote attackers to execute arbitrary commands.

IAV:

BugtraqID:

529 (http://www.securityfocus.com/bid/529)
- NT IIS MDAC RDS Vulnerability

Affected Machines:

IP Address

Machine Name

OS

☐

192.168.002.201

unknown

Microsoft Windows NT 4.0 SP3, Microsoft Windows NT 4.0 Server SP5-SP6, Microsoft Windows 2000 Professional RC1 or Windows 2000 Advanced Server Beta3

Retina - Network Security Scanner
Network Vulnerability Assessment & Remediation Management

Thursday, March 17, 2005

Sendmail 8.12.9 Buffer Overflow

Risk Level:

High

Category:

Mail Servers

Description:

The Sendmail 8.12.9 prescan function in Sendmail 8.12.9 allows remote attackers to execute arbitrary code via buffer overflow attacks.

How To Fix:

Upgrade to the latest version of Sendmail immediately.

Related Links:

Bugtraq posting (http://marc.theaimsgroup.com/?l=bugtraq&m=106381604923204&w=2)

CVE:

CAN-2003-0694 (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2003-0694)
- The prescan function in Sendmail 8.12.9 allows remote attackers to execute arbitrary code via buffer overflow attacks, as demonstrated using the parseaddr function in parseaddr.c.

IAV:

BugtraqID:

8641 (http://www.securityfocus.com/bid/8641)
- Sendmail Prescan() Variant Remote Buffer Overrun Vulnerability

Affected Machines:

IP Address

Machine Name

OS

☐

192.168.002.202

unknown

Linux 2.4.18 - 2.4.20 (X86)

Retina - Network Security Scanner
Network Vulnerability Assessment & Remediation Management

Thursday, March 17, 2005

Sendmail address field parsing buffer overflow

Risk Level:

High

Category:

Mail Servers

Description:

Sendmail 8.12.7 and earlier contains a flaw in its message header address field parsing routine that can be leveraged to cause a buffer overflow. A remote attacker can exploit this vulnerability, using a specially-crafted "From", "To", or "CC" header, to execute arbitrary code in the context of the sendmail daemon.

How To Fix:

Upgrade to the most current version of Sendmail, or apply the appropriate vendor-provided patch.

Related Links:

CERT Advisory CA-2003-07 (http://www.cert.org/advisories/CA-2003-07.html)
Sendmail Consortium home page (http://www.sendmail.org/)

CVE:

CVE-2002-1337 (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2002-1337)
 - Buffer overflow in Sendmail 5.79 to 8.12.7 allows remote attackers to execute arbitrary code via certain formatted address fields, related to sender and recipient header comments as processed by the crackaddr function of headers.c.

IAV:

BugtraqID:

6991 (http://www.securityfocus.com/bid/6991)
 - Sendmail Header Processing Buffer Overflow Vulnerability

Affected Machines:

IP Address
Machine Name
OS

☐
192.168.002.202
unknown
Linux 2.4.18 - 2.4.20 (X86)

Retina - Network Security Scanner
Network Vulnerability Assessment & Remediation Management

Thursday, March 17, 2005

Sendmail DNS Map TXT Overflow

Risk Level:

High

Category:

Mail Servers

Description:

A remotely exploitable buffer overflow exists in Sendmail versions 8.11 through 8.12.4. This vulnerability only exhibits itself if you have modified the configuration file to look up TXT records in DNS.

How To Fix:

Upgrade to the latest version of Sendmail.

Related Links:

Sendmail Consortium home page (http://www.sendmail.org/)

CVE:

CVE-2002-0906 (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2002-0906)
- Buffer overflow in Sendmail before 8.12.5, when configured to use a custom DNS map to query TXT records, allows remote attackers to cause a denial of service and possibly execute arbitrary code via a malicious DNS server.

IAV:

BugtraqID:

5122 (http://www.securityfocus.com/bid/5122)
- Sendmail DNS Map TXT Record Buffer Overflow Vulnerability

Affected Machines:

IP Address

Machine Name

OS

☐

192.168.002.202

unknown

Linux 2.4.18 - 2.4.20 (X86)

Retina - Network Security Scanner
Network Vulnerability Assessment & Remediation Management

Thursday, March 17, 2005

Sendmail prescan() address buffer overflow

Risk Level:

High

Category:

Mail Servers

Description:

Sendmail 8.12.8 and earlier contains a buffer overflow vulnerability in its handling of e-mail addresses that can be precipitated by the use of a special character value. An attacker can exploit this vulnerability to execute arbitrary code in the context of the mail server.

How To Fix:

Upgrade to the most current version of Sendmail, or apply the appropriate vendor-supplied patch.

Related Links:

CERT Advisory CA-2003-12 (http://www.cert.org/advisories/CA-2003-12.html)
Sendmail Consortium home page (http://www.sendmail.org/)

CVE:

CAN-2003-0161 (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2003-0161)
- The prescan() function in the address parser (parseaddr.c) in Sendmail before 8.12.9 does not properly handle certain conversions from char and int types, which can cause a length check to be disabled when Sendmail misinterprets an input value as a special "NOCHAR" control value, allowing attackers to cause a denial of service and possibly execute arbitrary code via a buffer overflow attack using messages, a different vulnerability than CAN-2002-1337.

IAV:

BugtraqID:

7230 (http://www.securityfocus.com/bid/7230)
- Sendmail Address Prescan Memory Corruption Vulnerability

Affected Machines:

IP Address
Machine Name
OS

☐
192.168.002.202
unknown

Retina - Network Security Scanner
Network Vulnerability Assessment & Remediation Management

Thursday, March 17, 2005

wu-ftp 2.6.1 format string when debug set

Risk Level:

High

Category:

FTP Servers

Description:

A format string class vulnerability in wu-ftp 2.6.1 and earlier, when running with debug mode enabled, allows remote attackers to execute arbitrary commands via a malformed argument that is recorded in a PASV port assignment.

How To Fix:

Upgrading wuftpd to the latest version will eliminate this, and other vulnerabilities discovered in the past. Otherwise maks sure wuftpd isn't be launched with the flags -d or -v.

Related Links:

WU-FTPD Development Group (http://www.wu-ftpd.org/)

CVE:

CVE-2001-0187 (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2001-0187)
- Format string vulnerability in wu-ftp 2.6.1 and earlier, when running with debug mode enabled, allows remote attackers to execute arbitrary commands via a malformed argument that is recorded in a PASV port assignment.

IAV:

BugtraqID:

2296 (http://www.securityfocus.com/bid/2296)
- Wu-Ftpd Debug Mode Client Hostname Format String Vulnerability

Affected Machines:

IP Address

Machine Name

OS

☐

192.168.002.202
unknown
Linux 2.4.18 - 2.4.20 (X86)

Retina - Network Security Scanner
Network Vulnerability Assessment & Remediation Management

Thursday, March 17, 2005

wu-ftpd fb_realpath() Off-By-One Buffer Overflow

Risk Level:

High

Category:

FTP Servers

Description:

A vulnerability exists within wu-ftpd, which affects the implementation of realpath(). This may allow for an attacker to execute arbitrary code in order to obtain root privileges.

How To Fix:

Apply the appropriate patch from the vendor.

Related Links:

Vendor provided patch (ftp://ftp.wu-ftpd.org/pub/wu-ftpd/patches/apply_to_2.6.2/realpath.patch)
Red Hat Advisory (https://rhn.redhat.com/errata/RHSA-2003-245.html)

CVE:

CAN-2003-0466 (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2003-0466)
- Off-by-one error in the fb_realpath() function, as derived from the realpath function in BSD, may allow attackers to execute arbitrary code, as demonstrated in wu-ftpd 2.5.0 through 2.6.2 via commands that cause pathnames of length MAXPATHLEN+1 to trigger a buffer overflow, including (1) STOR, (2) RETR, (3) APPE, (4) DELE, (5) MKD, (6) RMD, (7) STOU, or (8) RNTO.

IAV:

BugtraqID:

8315 (http://www.securityfocus.com/bid/8315)
- Multiple Vendor C Library realpath() Off-By-One Buffer Overflow Vulnerability

Affected Machines:

IP Address
Machine Name
OS

☐
192.168.002.202
unknown
Linux 2.4.18 - 2.4.20 (X86)

Retina - Network Security Scanner
Network Vulnerability Assessment & Remediation Management

Thursday, March 17, 2005

wu-ftpd File Globbing Vulnerability

Risk Level:

High

Category:

FTP Servers

Description:

Wu-Ftpd allows for clients to organize files for ftp actions based on file globbing patterns.The implementation of file globbing included in Wu-Ftpd contains a heap corruption vulnerability that may allow for an attacker to gain remote root access.

How To Fix:

Contact your vendor or visit their website to obtain a fix or software upgrade to eliminate this vulnerability.

Related Links:

CA-2001-33: Multiple Vulnerabilities in WU-FTPD (http://securityfocus.com/advisories/3701)
Wu-Ftpd Homepage (http://www.wu-ftpd.org/)

CVE:

CVE-2001-0550 (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2001-0550)
- wu-ftpd 2.6.1 allows remote attackers to execute arbitrary commands via a "~{" argument to commands such as CWD, which is not properly handled by the glob function (ftpglob).

IAV:

BugtraqID:

3581 (http://www.securityfocus.com/bid/3581)
- Wu-Ftpd File Globbing Heap Corruption Vulnerability

Affected Machines:

IP Address
Machine Name
OS

☐
192.168.002.202
unknown
Linux 2.4.18 - 2.4.20 (X86)

Retina - Network Security Scanner
Network Vulnerability Assessment & Remediation Management

Thursday, March 17, 2005

wu-ftpd globbing buffer overflow vulnerability

Risk Level:

High

Category:

FTP Servers

Description:

wu-ftpd 2.6.1 and earlier is vulnerable to a buffer overflow in the portion of ftpglob() matching open and close brackets, that can lead to user-supplied data being passed to the free() function. This condition can be exploited to execute arbitrary code.

How To Fix:

Upgrade to the most current version of wu-ftpd to eliminate this and possibly other security vulnerabilities in the product.

Related Links:

WU-FTPD Development Group home page (http://www.wu-ftpd.org/)

CVE:

CAN-2001-0935 (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2001-0935)
- Vulnerability in wu-ftpd 2.6.0, and possibly earlier versions, which is unrelated to the ftpglob bug described in CAN-2001-0550.

IAV:

Affected Machines:

IP Address
Machine Name
OS

☐
192.168.002.202
unknown
Linux 2.4.18 - 2.4.20 (X86)

Retina - Network Security Scanner
Network Vulnerability Assessment & Remediation Management

Thursday, March 17, 2005

wu-ftpd setproctitle() format string vulnerability

Risk Level:

High

Category:

FTP Servers

Description:

wu-ftpd 2.6.0 and earlier contains a format string vulnerability in its call to set_proc_title() that may allow a remote attacker to cause a denial of service or possibly execute arbitrary code.

How To Fix:

Upgrade to the most current version of wu-ftpd to eliminate this and possibly other security vulnerabilities in the software.

Related Links:

WU-FTPD Development Group home page (http://www.wu-ftpd.org/)

CVE:

CAN-2000-0574 (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2000-0574)
- FTP servers such as OpenBSD ftpd, NetBSD ftpd, ProFTPd and Opieftpd do not properly cleanse untrusted format strings that are used in the setproctitle function (sometimes called by set_proc_title), which allows remote attackers to cause a denial of service or execute arbitrary commands.

IAV:

BugtraqID:

1425 (http://www.securityfocus.com/bid/1425)
- Multiple Vendor ftpd setproctitle() Format String Vulnerability

Affected Machines:

IP Address

Machine Name
OS

☐
192.168.002.202
unknown
Linux 2.4.18 - 2.4.20 (X86)

Retina - Network Security Scanner
Network Vulnerability Assessment & Remediation Management

Thursday, March 17, 2005

wu-ftpd v2.6.0 conversion

Risk Level:

High

Category:

FTP Servers

Description:

A vunerability was found in wu-ftpd 2.6.0 and earlier that allows a remote attacker to gain root access to any wu-ftpd server that offers the conversion service. The attack works by uploading filenames with dashes that appear to be tar archives.

How To Fix:

Upgrading to the most recent version of wu-ftpd will correct this and other serious security vulnerabilities that have been found in 2.6.0.

Related Links:

WU-FTPD Development Group (http://www.wu-ftpd.org)

CVE:

CVE-1999-0997 (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0997)
- wu-ftp with FTP conversion enabled allows an attacker to execute commands via a malformed file name that is interpreted as an argument to the program that does the conversion, e.g. tar or uncompress.

IAV:

BugtraqID:

2240 (http://www.securityfocus.com/bid/2240)
- Multiple Vendor FTP Conversion Vulnerability

Affected Machines:

IP Address

Machine Name

OS

☐

192.168.002.202

unknown

Linux 2.4.18 - 2.4.20 (X86)

Retina - Network Security Scanner
Network Vulnerability Assessment & Remediation Management

Thursday, March 17, 2005

Anonymous FTP

Risk Level:

Medium

Category:

FTP Servers

Description:

It is recommended that you disable anonymous FTP access if it is not needed. Anonymous FTP access can lead to an attacker gaining information about your system that can possibly lead to them gaining access to your system.

How To Fix:

Follow your FTP server instructions on how to disable anonymous FTP.

CVE:

CAN-1999-0497 (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0497)
- Anonymous FTP is enabled

IAV:

Affected Machines:

IP Address

Machine Name

OS

☐

192.168.002.202

unknown

Linux 2.4.18 - 2.4.20 (X86)

- TCP:21

Retina - Network Security Scanner
Network Vulnerability Assessment & Remediation Management

Thursday, March 17, 2005

IIS Cumulative - ASP Server-Side Include Overflow

Risk Level:

Medium

Category:

Web Servers

Description:

There exists a local buffer overflow within how IIS handles ASP server side include variables. Attackers can leverage this vulnerability to execute malicious code.

How To Fix:

Install the appropriate Microsoft patch.

Related Links:

Microsoft Knowledge Base Article Q319733 (http://support.microsoft.com/default.aspx?scid=kb;en-us;Q319733)
Microsoft Security Bulletin MS02-018 (http://www.microsoft.com/technet/security/bulletin/MS02-018.mspx)

CVE:

CVE-2002-0149 (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2002-0149)
- Buffer overflow in ASP Server-Side Include Function in IIS 4.0, 5.0 and 5.1 allows remote attackers to cause a denial of service and possibly execute arbitrary code via long file names.

IAV:

BugtraqID:

4478 (http://www.securityfocus.com/bid/4478)
- Microsoft IIS ASP Server-Side Include Buffer Overflow Vulnerability

Affected Machines:

IP Address

Machine Name

OS

☐
192.168.002.201
unknown
Microsoft Windows NT 4.0 SP3, Microsoft Windows NT 4.0 Server SP5-SP6, Microsoft Windows 2000 Professional RC1 or Windows 2000 Advanced Server Beta3

Retina - Network Security Scanner
Network Vulnerability Assessment & Remediation Management

Thursday, March 17, 2005

IIS Cumulative - IIS Help File search flaw

Risk Level:

Medium

Category:

Web Servers

Description:

There exists a cross site scripting vulnerability within Microsoft IIS's help file search functionality. It can be used by attackers to perform malicious activities against other users of your web server.

How To Fix:

Install the appropriate Microsoft patch.

Related Links:

Microsoft Knowledge Base Article Q319733 (http://support.microsoft.com/default.aspx?scid=kb;en-us;Q319733)
Microsoft Security Bulletin MS02-018 (http://www.microsoft.com/technet/security/bulletin/MS02-018.mspx)

CVE:

CVE-2002-0074 (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2002-0074)
- Cross-site scripting vulnerability in Help File search facility for Internet Information Server (IIS) 4.0, 5.0 and 5.1 allows remote attackers to embed scripts into another user's session.

IAV:

BugtraqID:

4483 (http://www.securityfocus.com/bid/4483)
- Microsoft IIS Help File Search Cross Site Scripting Vulnerability

Affected Machines:

IP Address

Machine Name

OS

☐
192.168.002.201
unknown
Microsoft Windows NT 4.0 SP3, Microsoft Windows NT 4.0 Server SP5-SP6, Microsoft Windows 2000 Professional RC1 or Windows 2000 Advanced Server Beta3

Retina - Network Security Scanner
Network Vulnerability Assessment & Remediation Management

Thursday, March 17, 2005

IIS Cumulative - URL error handling bug

Risk Level:

Medium

Category:

Web Servers

Description:

There exists a bug within Microsoft IIS by which it's possible to launch a denial of service attack against the web server through the use of null characters.

How To Fix:

Install the appropriate Microsoft patch.

Related Links:

Microsoft Knowledge Base Article Q319733 (http://support.microsoft.com/default.aspx?scid=kb;en-us;Q319733)
Microsoft Security Bulletin MS02-018 (http://www.microsoft.com/technet/security/bulletin/MS02-018.mspx)

CVE:

CVE-2002-0072 (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2002-0072)
- The w3svc.dll ISAPI filter in Front Page Server Extensions and ASP.NET for Internet Information Server (IIS) 4.0, 5.0, and 5.1 does not properly handle the error condition when a long URL is provided, which allows remote attackers to cause a denial of service (crash) when the URL parser accesses a null pointer.

IAV:

BugtraqID:

4479 (http://www.securityfocus.com/bid/4479)
- Microsoft IIS ISAPI Filter Access Violation Denial of Service Vulnerability

Affected Machines:

IP Address

Machine Name

OS

☐

192.168.002.201

unknown

Microsoft Windows NT 4.0 SP3, Microsoft Windows NT 4.0 Server SP5-SP6, Microsoft Windows 2000 Professional RC1 or Windows 2000 Advanced Server Beta3

Retina - Network Security Scanner
Network Vulnerability Assessment & Remediation Management

Thursday, March 17, 2005

telnet service

Risk Level:

Medium

Category:

Remote Access

Description:

Telnet is a service that allows a remote user to connect to a machine. Telnet sends all usernames, passwords, and data unencrypted.

How To Fix:

Consult your user manual or help file for information on how to disable your telnet service. If no user manual or help file exists then contact your software vendor.

CVE:

CAN-1999-0619 (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0619)
- The Telnet service is running.

IAV:

Affected Machines:

IP Address
Machine Name
OS

☐
192.168.002.203
unknown
Cisco IOS 11.3 - 12.0(11)

- TCP:23

Retina - Network Security Scanner
Network Vulnerability Assessment & Remediation Management

Thursday, March 17, 2005

wu-ftpd globbing system resource exhaustion

Risk Level:

Medium

Category:

FTP Servers

Description:

wu-ftpd 2.6.0 and earlier is susceptible to a denial-of-service condition due to its globbing implementation. If a user without resource usage restrictions submits a specially-crafted globbing string (e.g., /../*/../*), he can exhaust system resources.

How To Fix:

Upgrade to the most current version of wu-ftpd to eliminate this and possibly other security vulnerabilities in the software.

Related Links:

WU-FTPD Development Group home page (http://www.wu-ftpd.org/)

IAV:

BugtraqID:

2496 (http://www.securityfocus.com/bid/2496)
- Multiple Vendor FTP glob Expansion Vulnerability

Affected Machines:

IP Address

Machine Name

OS

☐

192.168.002.202

unknown

Linux 2.4.18 - 2.4.20 (X86)

Retina - Network Security Scanner
Network Vulnerability Assessment & Remediation Management

Thursday, March 17, 2005

wu-ftpd privatepw symbolic link file overwriting

Risk Level:

Medium

Category:

FTP Servers

Description:

wu-ftpd 2.6.0 and 2.6.1 contain a race condition in its privatepw group access configuration utility that allows an attacker to corrupt arbitrary files, by creating symbolic links with the names of temporary files likely to be used by privatepw.

How To Fix:

Upgrade to the most current version of wu-ftpd to eliminate this and possibly other security vulnerabilities in the software.

Related Links:

WU-FTPD Development Group home page (http://www.wu-ftpd.org/)

CVE:

CVE-2001-0138 (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2001-0138)
- privatepw program in wu-ftpd before 2.6.1-6 allows local users to overwrite arbitrary files via a symlink attack.

IAV:

BugtraqID:

2189 (http://www.securityfocus.com/bid/2189)
- wu-ftpd /tmp File Race Condition Vulnerability

Affected Machines:

IP Address

Machine Name

OS

☐
192.168.002.202
unknown
Linux 2.4.18 - 2.4.20 (X86)

Retina - Network Security Scanner
Network Vulnerability Assessment & Remediation Management

Thursday, March 17, 2005

IIS Cumulative - XSS flaw in HTTP Error

Risk Level:

Low

Category:

Web Servers

Description:

There exists a cross site scripting flaw in how IIS handles HTTP errors. It can be used to trick users into viewing malicious website content as though it were hosted on the susceptible server.

How To Fix:

Install the appropriate Microsoft patch.

Related Links:

Microsoft Knowledge Base Article Q319733 (http://support.microsoft.com/default.aspx?scid=kb;en-us;Q319733)
Microsoft Security Bulletin MS02-018 (http://www.microsoft.com/technet/security/bulletin/MS02-018.mspx)

CVE:

CVE-2002-0148 (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2002-0148)
- Cross-site scripting vulnerability in Internet Information Server (IIS) 4.0, 5.0 and 5.1 allows remote attackers to execute arbitrary script as other users via an HTTP error page.

IAV:

BugtraqID:

4486 (http://www.securityfocus.com/bid/4486)
- Microsoft IIS HTTP Error Page Cross Site Scripting Vulnerability

Affected Machines:

IP Address

Machine Name

OS

☐
192.168.002.201
unknown
Microsoft Windows NT 4.0 SP3, Microsoft Windows NT 4.0 Server SP5-SP6, Microsoft Windows 2000 Professional RC1 or Windows 2000 Advanced Server Beta3

Retina - Network Security Scanner
Network Vulnerability Assessment & Remediation Management

Thursday, March 17, 2005

IIS Cumulative - XSS in Redirect Response

Risk Level:

Low

Category:

Web Servers

Description:

There exists a cross site scripting vulnerability within Microsoft IIS. It can allow an attacker to make web visitors view malicious content as though it were hosted on the susceptible web server.

How To Fix:

Install the appropriate Microsoft patch.

Related Links:

Microsoft Knowledge Base Article Q319733 (http://support.microsoft.com/default.aspx?scid=kb;en-us;Q319733)
Microsoft Security Bulletin MS02-018 (http://www.microsoft.com/technet/security/bulletin/MS02-018.mspx)

CVE:

CVE-2002-0075 (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2002-0075)
- Cross-site scripting vulnerability for Internet Information Server (IIS) 4.0, 5.0 and 5.1 allows remote attackers to execute arbitrary script as other web users via the error message used in a URL redirect (""302 Object Moved") message.

IAV:

BugtraqID:

4487 (http://www.securityfocus.com/bid/4487)
- Microsoft IIS HTTP Redirect Cross Site Scripting Vulnerability

Affected Machines:

IP Address

Machine Name

OS

☐

192.168.002.201

unknown

Microsoft Windows NT 4.0 SP3, Microsoft Windows NT 4.0 Server SP5-SP6, Microsoft Windows 2000 Professional RC1 or Windows 2000 Advanced Server Beta3

Retina - Network Security Scanner
Network Vulnerability Assessment & Remediation Management

Thursday, March 17, 2005

HTTP TRACE method supported

Risk Level:
Information

Category:
Web Servers

Description:
Retina has discovered that the target host supports the HTTP TRACE method.

Related Links:
CERT Vulnerability Note (http://www.kb.cert.org/vuls/id/867593)

IAV:

Affected Machines:

IP Address
Machine Name
OS

☐
192.168.002.201
unknown
Microsoft Windows NT 4.0 SP3, Microsoft Windows NT 4.0 Server SP5-SP6, Microsoft Windows 2000 Professional RC1 or Windows 2000 Advanced Server Beta3

- TCP:80

Retina - Network Security Scanner
Network Vulnerability Assessment & Remediation Management

Thursday, March 17, 2005

No Remote Registry Access Available

Risk Level:

Information

Category:

Registry

Description:

This alert is only to notify you that Retina was not able to access the remote system's registry. Without registry access, Retina will still be able to remotely audit for vulnerabilities, although having access to the remote registry does provide Retina with the ability to verify if specific security patches are installed.

Retina only uses the credentials of the account under which it is executed when accessing the remote system's registry -- it will not use any "net use" supplied credentials. Therefore, we recommend executing Retina as a domain administrator account.

How To Fix:

Ensure that the system has remote registry capabilities enabled, and that you have administrative rights on the system.

IAV:

Affected Machines:

IP Address
Machine Name
OS

☐
192.168.002.201
unknown
Microsoft Windows NT 4.0 SP3, Microsoft Windows NT 4.0 Server SP5-SP6, Microsoft Windows 2000 Professional