Global Information Assurance Certification Paper

What's in your bag?

A Unix Security Auditors Toolkit.

By

Ian C. Ware

GSEC Security Essentials Certification
Practical Assignment: Version 1.4b
Option 1

11/1/2002

Table of Contents

Abstract

With the rise of computer intrusion incidents, it is becoming increasingly clear that companies need to be more cyber-aware, meaning solely that they need a clear understanding of their infrastructure, and the possible exploits that they are making available to potential attackers. It is the job of the systems security auditor to detect and document the system vulnerabilities for their clients. As with any technician, beyond their individual skill and prowess, a very important part of the job is the tools that are used to find the exploits, so that they may be presented to the client so that they may better prepare their enterprise against the increasing threats of cyberterrorism, and the ubiquitous hackers. This paper will discuss a few tools available to the security auditor, to aid them in their assessment of a computer system and/or network.

Introduction

In 1988, I believe that I hardly ever heard the word terrorism, let alone *cyberterrorism*. The FBI defines terrorism as "the unlawful use of force or violence against persons or property to intimidate or coerce a government, the civilian population, or any segment thereof, in furtherance of political or social objectives."[1] C*yberterrorism*, on the other hand, takes the definition further. While as of this date there appears to be no clear-cut definition of cyberterrorism, I take it to mean the usage of digital means, in the pursuance of terrorist activities, or as a direct terrorist attack. The number of computer attacks reported in 1988 was hardly worth mentioning. According to the Computer Emergency Response Team's (CERT) website hosted by Carnegie Mellon, the number of incidents handled in 1988 was only 6. Compare that to the figures posted at the end of 3Q02, an astounding 73,359 incidents[2], and with three months left in the year. In Steve Rangers recent article "Government fights off 6,000 online attacks", he states that in October alone, the British government had to fight off more than 6500 computer attacks this year, and that the Cabinet office was hit with almost 1,200 alone in the month of September. Fortunately, according to one cabinet minister, none of the attacks had resulted in any compromised systems, or damage to the information held within[3].

The job of a security auditor is to help ensure that computer and network resources are less vulnerable to attack. Note, I did not say impervious. The only way to truly make a system impervious is to have it sitting in a locked room, with guards at the door, and turned off. Once the ability for anyone, including an authorized individual, gains access to a system, it is vulnerable to attack. As a systems security auditor, I spent a great deal of time trying to figure out the best

[1] Code of Federal Regulations, Title 28 Volume1 Section 0.85
[2] CERT/CC Statistics, http://www.cert.org/stats
[3] Ranger, Steve. http://www.vnunet.com/News/1137117

tools to assess the vulnerabilities of a system. Fortunately, there are a great number of them to choose from. The most important things to consider in choosing a tool are; a) does it provide you the functionality you need, b) is it configurable and extensible, and c) is it supported. As primarily a UNIX consultant, all of my tools will be either UNIX or LINUX based. Most if not all, are available as either downloadable source or precompiled binaries for a multitude of operating systems. Many are also available as Windows based tools, but I have heard of some issues either in the compilation or installation of some, on a windows platform.

The discussion will be broken up into to main categories, Tools and Auditing. The tools section will discuss just that, the tools themselves. While there are many other tools available, I will discuss the ones that I have chosen to use over my career. These include for network scanning tcpdump, ethereal, and kismet. My system scanning tools are nmap, nessus, and a collection of scripts and command lines that I can run on a particular host. Finally, I use a set of password auditing tools such as dsniff and John the Ripper (here forward referred to only as john).

The auditing section will discuss the use of the tools to provide an audit of a network, host, or collection of hosts, and a methodology for creating the documentation from the output.

Tools

As with the tools of any craftsman, a security auditor should pick the best tool that he feels meets his need. Additionally, like a woodworker, his tools should be continually sharpened, and ready for use. In the world of computers, this means that your operating system as well as your products should be up to date on the latest security patches, and continually reviewed for bugs and other exploits. The hardware that one uses needs to be maintained for firmware updates, as there are exploits designed to attack network cards and systems bios that are not updated.

Patches are usually available from both the OS vendor, and the hardware vendor. Notification of these can generally be made through security mailing lists from the OS vendor. Additionally, security advisory mailing lists from the local incident response team provide a wealth of knowledge and information on up to date issues[4].

Laptop

One of the most important tools is the auditor's laptop from which he may run the various scans, and document the results. In this particular instance, I don't wholly

[4] UNIX Security Checklist

include just the hardware, but I also speak to the Operating System installed. What I find critical in this instance, along with all the other tools, is that you know a) how it is configured, b) how to fix the OS when problems are presented, and c) know how to update it. I prefer a UNIX based operating system, mainly due to the fact that I have more tools available to me, and that I work primarily on UNIX systems. This way, there are fewer chances of interoperability issues between my system, and those that I am auditing. Additionally, this allows me to write and test customized shell scripts, prior to running them on the audited host.

My operating system of choice is SuSE Linux Professional Edition. As of this writing, the current release is 8.1. One plus to the SuSE distribution is that most of the tools discussed, have a compiled version provided on the SuSE media, and patches are readily available via ftp server. The only downside that I have found to date, is that of the Linux distributions, SuSE appears to be the only one that does not provide the full operating system as a free ISO download.

I believe a critical step prior to assessing a client's network, especially if they believe they may have had an intrusion, is to ensure that my laptop is locked down and secured. If they believe they have had an attacker, then the last thing you would want would be someone to gain access to the tools or reports you are generating that might show additional exploits. However, since this paper is on the systems auditing, and not forensic analysis, the discussion of such is outside the scope of this paper.

The choice of laptop vendor is purely personal. I have no preferences one way or the other to any of the laptop vendors. I would, however, say that the laptop should have as much disk space and memory as you can afford. Additionally, once you begin running password cracking tools, you will want as much CPU power as possible. Keep in mind, if you can get it, so can the hackers. As a minimum for peripherals, ensure that you have at least a modem, 10/100 Ethernet adapter, and an 802.11 wireless network adapter if you ever intend on scanning wireless networks. Lastly, before you begin any auditing, make sure your system works. There is nothing worse than getting to a client site and having to troubleshoot networking, or compilation issues. This isn't to say that some issues may arise while on the client site, but that is why it is best to know your system and be able to comfortably support the installation.

Network Scanning Tools

Tcpdump

Tcpdump is a packet sniffer that is distributed with most versions of Linux. It is also available from http://www.tcpdump.org. The version provided with SuSE 8.1 is 3.7.1-94, with patches available that bring it up to 3.7.1-112. It is a very simple program that allows the pulling of traffic directly off the wire, and can write to a

file for later review[5]. The raw format that tcpdump outputs, can also be viewed in other packages, such as ethereal, to be mentioned below.

What is nice about tcpdump is that it is configurable to only capture certain packet types and can filter on any piece of the header in IP, UDP, ICMP, or TCP packets. Tcpdump, without any filtering or command line switches can provide important information such as:

- Source host
- Destination host
- Packet length
- Protocol
- Port numbers
- TCP sequence numbers (on TCP based packets)[6]

The downside is that this information is also usable by a hacker for purposes of session hijacking buy tracking and intercepting the TCP sequence numbers, but it is still a useful tool to analyze the current traffic, and possibly point out any anomalies that might exist such as abnormally high amounts of ACK packets. When a session is highjacked, and the attacker has neither guessed the right sequence number or not taken the client system offline, the server will begin to receive the packets from the attacker thinking they are from the legitimate client, however the sequence numbers may be out of order. In an attempt to re-sequence, the server will send out numerous SYN and ACK packets to which the client replies with its SYN and ACK packets. The result is an ACK storm, that will cause network degradation, but if you are watching with a tool such as tcpdump, the attack should be readily visible[7].

It is important, however, that prior to doing any packet sniffing with any tool, that a) you have permission, and b) you understand the traffic that is already supposed to be going across the network.

Ethereal

Ethereal is a freely available network packet sniffer and protocol analyzer that is available for many flavors of UNIX as well as Windows.

This is another tool that is provided with the SuSE distribution. Precompiled binaries, source code, and documentation are also available from http://www.ethereal.com. As of this writing, the version available from the SuSE distribution is 0.9.6-29, however from the Ethereal web site, version 0.9.7 is available.

---

[5] Tcpdump, http://www.iepm.slac.stanford.edu/monitoring/passive/tcpdump.html.
[6] Tcpdump
[7] Cole, Eric. Hacker Beware. pp. 151-152.

Features that make Ethereal such an appealing tool is:

- Data can either be read directly off the wire, or from captured data contained in a file.
- File data can be in a multitude of scan formats, mainly including tcpdump format which allows you to capture packets in a variety of ways, without the GUI and them analyze them later
- GUI has built in customizable filters
- Supports live capture from a variety of media including 802.11, Ethernet, FDDI, PPP, and others.
- Currently supports dissection of over 300 different protocols.
- TCP packets can be compiled and the ASCII output viewed as a contiguous sequence[8].

The tool functions well to see what kind of traffic is proliferating on your network, but also serves as a good troubleshooting tool for packet traces.

Kismet

Kismet is actually a tool for scanning wireless networks, or should I say, sniffing them out. The current stable version, as of this writing, is 2.6.2. and is available from http://www.kismetwireless.net as source code.

In today's world of always on, always-convenient Internet access, Wireless Access Points or WAPs are popping up everywhere. Many companies are looking at wireless LANs or WLANs as a low cost solution for the extension of their network perimeter and provide mobile access to their users. However, many companies are seeing access points pop up where they were not supposed to be.

A hacker can quite easily get onto an unsecured WAP by using a method called wardriving. Wardriving is simply walking or driving around with a laptop, a wireless NIC, and sniffing software, such as kismet, and finding open access points. Additionally, should one choose, kismet and others support GPS devices. This enables a user to not only seek out the wireless networks, but also create a real map of where they are for later use[9].

In its use, kismet will not only detail the IP ranges of the wireless networks, but will also provide ESSID or Extended Service Set Identifier. The ESSID identifies the wireless LAN and must match between client and access point for a connection to be made.

[8] Ethereal Network Analyzer. http://www.ethereal.com.
[9] Kismet. http://www.kismetwireless.com

System Scanning Tools

Nmap

Nmap, along with the following product, Nessus, are two of my favorite tools. Both are touted system port and/or vulnerability scanners, but they can do so much more. Bundled with the 8.1 SuSE release, nmap comes at version 3.00-53, but is also available from http://www.insecure.org/nmap/ which also currently has version 3.00. Nmap is one of those tools that you like to have around, just like your old trusty Swiss army knife. It is a simple command line utility that can perform such tasks as:

- Network scans for available hosts
- System scans for available ports
- OS Fingerprinting[10]

From the command line, nmap is highly customizable, just like tcpdump. My command line typically looks like:

```
# nmap -sT -I -O -p 1-10000 -v <list of servers>
```

Table 1: nmap options taken from nmap man page

-sT	Do a full TCP connect() scan. Other scans available so as to either reduce traffic, or reduce detection, if doing penetration testing, are: -sS TCP SYN scan [half open scan] -sF Stealth FIN -sX Xmas Tree -sN Null Scan -sP Ping Scan -sU UDP Scan -sO IP Protocol Scan -sA ACK Scan -sW Window Scan -sR RPC Scan -sL List scan
-I	Perform TCP reverse ident scanning on the open port thereby providing the owner of the process on the port.

[10] Nmap. http://www.insecure.org/nmap/index.html

-O	Perform OS fingerprinting
-p 1-10000	Scan the system on ports 1 – 10000. Can also make port selection non-contiguous, such as 1-90, 120, 125, 500-10000

Beyond the commands listed above, there is a wealth of other options to either make your information better, or aid in stealth, should you be performing penetration testing.

Nessus

In my opinion, Nessus is to system scanning as Arnold Schwarzenegger is to bodybuilding. Unlike many other security scanners, Nessus is quite extensible, and takes nothing for granted. It never considers the fact that services typically run on a fixed port. It will detect programs running on the ports, i.e. a web server running on port 2048, and then attempt to exploit the services[11]. The version shipped with SuSE 8.1 is 1.2.3-72 and comes packed with 1043 different plugins to run against a host system. Additionally, Nessus has a customized scripting language NASL (Nessus Attack Scripting Language), so that you may write your own plugins, and a utility to update the plugins from the central site[12].

Nessus has so many features that make it a phenomenal security scanner that to list them all would be far and away outside of the scope of this document. However, a highlight of what I find the be the most useful are:

- Client-server architecture – Can run the GUI on one system, and the attacks from the server side, on a separate system.
- Multithreaded attack – Can run scans on multiple systems at one time, depending upon the amount of power on the client side.
- Service Recognition – Can recognize applications running on non-standard ports, and then run the appropriate exploits to test the vulnerability.
- Non-destructive scans – You have the ability to turn off the exploits, which may potentially bring down the system.
- Smart Plugins – Nessus can optimize the scans by determining which exploits may or may not be usable against the host i.e. not running Apache exploits on an IIS server.
- Reporting – Nessus has incredible reporting features. The ability is there to change the level of reporting from minimal to severe, and with that, one not only learns the exploits, but many times, what it can take to fix it. The HTML output, by Nessus, includes links to the CVE (Common

[11] Nessus. http://www.nessus.org.
[12] Nessus.

Vulnerabilities and Exposures) repository providing detailed information on the vulnerability[13].

A little more on Nessus' reporting capability. Not only can it output the results in HTML format which includes the CVE links, and pretty nice pie graphs and charts, but can also export the reports as LaTeX, plain HTML, and plain ASCII text. This allows the reports to be easily parsed and processed for your particular documentation needs.

Personal Scripts

As far as personal scripts, one thing I maintain is a collection of one-line commands that can be used for information gathering.

- Finding SUID and SGID files
  # find / \(-perm -4000 -o -perm -2000 \) -exec ls -ld {} \;
- Finding globally writable directories
  # find / -type d -perm -002 -exec ls -ld {} \;
- Garnering SNMP information (if nmap shows SNMP running)
  # snmpget <host> public system.sysDescr.0
  # snmpwalk <host> public system

Additionally, I maintain OS specific scripts used to gather information about the system itself, software installed, along with revisions and patches, and hardware configured. Additionally, information such as users configured, those without passwords, as well as any potential root level accounts nefariously placed upon the system.

Password Auditing Tools

Dsniff

Simply put, dsniff is a network password sniffer. However, it ends up being more than that. Freshly compiled, dsniff supports the acquisition of passwords via such protocols as telnet, FTP, POP, AIM, HTTP, NNTP, IMAP, rlogin, LDAP, and a bunch of others. Additionally, it has the wonderful added feature of being able to adding addtional protocol fingerprints[14].

Dsniff does require a few other packages to be installed. Below are the requisite packages and from where they may be obtained:

- BerkelyDB – http://www.sleepycat.com/
- OpenSSL - http://www.openssl.org/
- Libpcap - http://www.tcpdump.org/

[13] Nessus.
[14] Dsniff. http://naughty.monkey.org/~dugsong/dsniff.

- Libnids – http://www.packetfactory.net/Projects/Libnids/
- Lbnet– http://www.packetfactory.net/Projects/Libnet/
- Fragrouter – http://www.wittys.com/toolfiles.html (not required, but useful)

What probably turns out to be the single most beneficial part of this package is, that when you run it, with the client watching, they can see all the clear text passwords that their people use, plus how many of them are very weak. Mind you this package also works extremely well when performing penetration tests. The dsniff source package also comes with another very important tool, arpspoof. With these tools, one may spoof the router, and perform sniffing across the switched networks. With arpspoof, you impersonate the local gateway[15]. However, unless you re-forward the packets on, a) you may not get anything, and b) you could take down the LAN. Neither of which would you want. Also, as with any of the scanning tools, make sure you first have permission to run the scan.

John

John is another one of those multipurpose tools that I like to have in my auditing arsenal. Where nmap is the Swiss Army knife for system and network scanning, John is the tool of my choice for password cracking. While other tools exist, such as the long time standard Crack, john is gaining popularity quickly because of its ease of installation, configuration, use, and its speed. Primarily, it was designed to detect weak UNIX passwords[16]. Currently, however it runs on, and has been tested with a number of UNIX flavors and Microsoft operating systems including DOS and many NT flavors. Not only does John provide you with a full-featured dictionary password cracker, but it also provides the ability to customize the attack schemes for a brute force attack. John allows you to create more complete dictionary files, for faster dictionary attacks and it even provides command line options, for running benchmark tests on a given system, to see how many characters per second it can actually run. Additional tools provided are:

- unshadow – Combines the password and shadow password files. This is output to standard out, so it must be run into a file for use by john.
- unafs – Retrieves the password hashes from the binary AFS database,.Again to standard out.
- unique – Removes the duplicates from a given wordlist that is provided from standard input.
- mailer – This shell script, with customizable message, sends mail to users who end up with weak passwords[17].

[15] Dsniff.
[16] John the Ripper. http://www.openwall.com/john
[17] John the Ripper

The version of John provided with my SuSE distribution is 1.6-531, however, as with all of the other tools documented here, John is available via RPM binary or compilable source from the home site listed in the references section at the end of this document. From a vanilla install, John is able to detect numerous ciphertext formats including DES-based, BSDI extended DES, many MD5-based, and Blowfish hashes. Also, with the use of one extra command to extract the passwords, it can crack AFS passwords and WinNT hashes[18].

Auditing

The Audit

When auditing for a client, to perform a thorough assessment, I usually start with a tcpdump scan, possibly in conjunction with ethereal. I want to see what kinds of traffic are actually running on the network. From there, I will run nmap scans on the servers and/or networks to see what ports are open, and derive the types of systems I'm dealing with. Granted, the customer usually provides this information, but for one thing, I like to make sure that the tools fingerprinting agrees with the client, and if it is different, I can get the signatures in order to update the nmap information.

Armed with the nmap information, I can compile a good list of hosts to scan using Nessus. Depending upon the client's wishes, I could either run the nessus scans during the day, or during off hours. There are a number of exploits nessus runs, which could potentially crash a system. Many clients would not like this to happen during the business day. Some also ask that those exploits not be run at all. If that is the case, I inform them that the assessment will not be totally complete without that information.

The rest of my time is spent editing and running custom scripts gathering information about the systems, and running john against the password files. If a password survives more than a few minutes, it is generally a pretty good password. To long, and an aggressor would probably give up rather than risk detection.

Documentation

The documentation from a security audit is probably the most important piece of the engagement. Without well-documented results, the actual audit would be worthless. One of my favorite things about UNIX is the ability to capture your session in a file using the script command, or using the Xterm logging function

[18] John the Ripper

found in most xterm interpretations. This has the benefit of being able to capture not only the output from your commands, but your commands as well. This makes the inclusion of such information into your final document, much easier.

I generally like to divide the assessment portion of my final document into the following sections:

- Environmental Overview
- Vulnerabilities
- Fixes
- Recommendations

This provides the client with all the pertinent information from my scans and assessments, plus some direction to address some of the issues.

Conclusion

The tools discussed here are not private, and by no means hard to get. Hackers use these tools on a daily basis for their own nefarious purposes, so it stands to reason that they could and should be used to help thwart those attacks. The tools carried by a security auditor are paramount in helping him find the customer's vulnerabilities. The ones I chose to discuss are the tools I use for auditing client's systems. However, It is up to each individual auditor himself to find the best tools that he knows the best, and that he feels can do the best job for him. He also needs to know how to manage and update those tools as new vulnerabilities are found, and patches possibly made available.

References

CERT/CC Statistics 1988-2002. 4 Oct. 2002. < http://www.cert.org/stats>.

Check Point Expands Leadership in Security Across
Mobile Devices and Wireless Networks.
<http://www.checkpoint.com/press/2002/wireless061802.html>.

Dsniff, <http://naughty.monkey.org/~dugsong/dsniff/>.

The Ethereal Network Analyzer. <http://www.ethereal.com>

Cole, Eric, Matthew Newfield, and John M. Millican. GSEC Security Essentials Toolkit. Que Publishing, 2002.

Cole, Eric. Hackers Beware. New Riders Publishing, 2002.

Ranger, Steve. "Government fights off 6,000 online attacks." vnunet.com. 26 Nov. 2002 <http://www.vnunet.com/News/1137117>.

John the Ripper. <http://www.openwall.com/john/>.

Kismet. <http://www.kismetwireless.com>.

Monitoring with tcpdump. <http://www-iepm.slac.stanford.edu/monitoring/passive/tcpdump.html>.

Nessus. <http://www.nessus.org/>.

Nmap. <http://www.insecure.org/nmap/index.html>.

Code of Federal Regulations, Title 28 Volume1. 28CFR0.85. Revised 1 July, 2002. < http://frwebgate3.access.gpo.gov/cgi-bin/waisgate.cgi?WAISdocID=6683652118+1+0+0&WAISaction=retrieve>

Tcpdump. <http://www.tcpdump.org>.

UNIX Security Checklist v2.0. <http://www.cert.org/tech_tips/usc20_full.html>