GIAC CERTIFICATIONS

Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permited without express written permission.

Interested in learning more?

Check out the list of upcoming events offering
"Security Essentials: Network, Endpoint, and Cloud (Security 401)"
at http://www.giac.org/registration/gsec

Vince Kornacki
GSEC Practical v.1.3
"Arming Linux With Penetration Testing Tools"

Introduction

This paper identifies and describes several useful penetration testing tools that can be run on Linux. Linux is a powerful, customizable, free operating system. After reading this paper, users should be able to configure a Linux workstation as an effective penetration-testing platform. Amazingly, every tool discussed in this paper is available free of charge. Many of the tools are governed by the GNU GPL (General Public License), which means that they can be improved and redistributed by other users. In addition to a description of *what* the tool does, this paper explains *why* the tool is useful. Where applicable, sanitized tool output is displayed.

The tools are organized by penetration testing phase. For example, port scanners are identified and described in the "Scanning" phase section. For the purposes of this paper, the phases of a penetration test are as follows:

- Target Identification
- Footprinting
- Scanning
- Exploiting

Before you use any of the tools described in this paper, it is extremely important that you first *ask for permission*. If you are using these tools on a client project, make sure the client has signed a contract. A contract is invaluable in the event of legal action. Likewise, the client will undoubtedly require you to sign an NDA (Non-Disclosure Agreement). These formalities may not be convenient, but they are extremely important since the laws surrounding penetration testing are very much a gray area.

The tools detailed in this document were all successfully compiled on Red Hat Linux 7.1, kernel 2.4.12. They should compile just as easily on other flavors of Linux. The current version of Red Hat Linux can be freely downloaded from http://www.redhat.com/products/ [1]. In addition, documentation is available at http://www.redhat.com/support/resources/install_upgrade/installing_linux.html [1].

Target Identification

The first phase of a penetration test is "Target Identification". The Target Identification phase is more complicated than simply identifying the name of your client. What exactly are you going to be penetration testing? An entire network? A specific segment on the network? A specific server on a segment? A specific application on a server? Or maybe something else, such as a specific router, firewall, or PBX? It is important to accurately identify what system(s) are being examined. If the penetration test is a client project, this information will most likely be defined in the contract.

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

In addition, some clients may be extremely upset if they think that you have gone "too far" on a penetration test. Consequently, it is extremely important to ensure that the contract explicitly defines what is acceptable and what is not acceptable. Are you allowed to compromise production servers? If so, should you attempt to download sensitive information such as credit card numbers? Are DoS (Denial of Service) attacks acceptable? Be sure to clear up any confusion *before* the penetration test begins.

Footprinting

The next phase of a penetration test is "Footprinting". During the Footprinting phase any and all applicable information regarding the target is located. The goal of the Footprinting phase is to build a database of information that can be used during the rest of the penetration test. There are several useful tools that can be utilized during the Footprinting phase:

- Whois

  The "whois" command is built into the Linux operating system. The whois command provides detailed information regarding an organization:

```
root@linux $ whois company.com
[whois.crsnic.net]

Whois Server Version 1.3

   Domain Name: COMPANY.COM
   Registrar: NETWORK SOLUTIONS, INC.
   Whois Server: whois.networksolutions.com
   Referral URL: http://www.networksolutions.com
   Name Server: NAMESERVER.COMPANY.COM
   Updated Date: 07-feb-2002

Registrant:
   Company, Inc. (COMPANY-DOM)
   123 Main Street
   Chicago, IL 60661
   US

   Administrative Contact, Technical Contact:
   John Doe (JD123-ORG) john.doe@company.com
   Company, Inc.
   123 Main Street
   Chicago, IL 60661
   US
   312-123-4567
   Fax- - 312-123-5678

   Record last updated on 07-Feb-2002.
   Record expires on 03-Jun-2011.
   Record created on 02-Jun-1995.
   Database last updated on 20-Mar-2002 05:30:00 EST.

   Domain servers in listed order:

   NAMESERVER.COMPANY.COM                10.1.1.1
```

© SANS Institute 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

The output of the whois command contains a wealth of useful information. For example, names listed as administrative and technical contacts can be used for social engineering attacks. While some organizations use a generic name like “Hostmaster”, others list the actual names of the administrative and technical contacts (see “John Doe” in the whois output above). This individual is most likely a knowledgeable member of the client’s IT department. Armed with this knowledge, an attacker could call the client’s help desk and impersonate this individual. Perhaps the attacker could ask the help desk to reset a password, or create a temporary account. Persuasive attackers can go a long way with just a little information.

In addition, phone numbers listed in the whois output can be used to launch wardialing attacks. Wardialing involves dialing a block of telephone numbers, searching for modems or PBXs. The listed phone numbers can provide attackers with area codes and exchanges to conduct a wardial. Using a wardialing tool, an attacker could search the (312) 123-XXXX exchange for modems (see “312-123-4567” in the whois output above).

Furthermore, DNS servers listed in the whois output can be further probed. By using the “dig” command, attackers can attempt zone transfers, or query the DNS servers for their exact BIND version. The next section provides further information regarding the dig command.

As an alternative to the whois command, websites such as the VeriSign Global Registry Services Whois webpage can be used to conduct whois searches [2]. These websites provide the same information as the whois command, but in a more attractive format.

- Dig

  Like whois, the ”dig” command is also built into the Linux operating system. The dig command can be used to query DNS servers and conduct zone transfers. If allowed, zone transfers allow unauthenticated users to download all DNS records for a specified domain:

```
root@linux $ dig @nameserver.company.com company.com axfr
; <<>> DiG 9.1.0 <<>> @nameserver.company.com company.com axfr
;; global options:  printcmd
company.com.                3600     IN       SOA      nameserver.company.com.
admin@company.com. 10       3600 3600 86400 3600
company.com.                3600     IN       NS       nameserver.company.com.
company.com.                3600     IN       MX       10 mailserver.company.com.
nameserver.company.com.     3600     IN       A        10.1.1.1
mailserver.company.com.     3600     IN       A        10.1.1.2
www.company.com.            3600     IN       A        10.1.1.3
;; Query time: 350 msec
;; SERVER: 10.1.1.1#53(nameserver.company.com)
;; WHEN: Wed Mar 20 15:20:58 2002
;; XFR size: 6 records
```

  In this case, the DNS server “nameserver.company.com” allowed a zone transfer for the “company.com” domain. Six records were returned. There are records of DNS servers (designated by NS), mail servers (designated by

© SANS Institute 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

MX), and hosts (designated by A). Armed with these addresses, attackers can launch port scans and vulnerability scans. Zone transfers may also reveal the existence of test servers, which typically contain "test" in the name. Test servers are usually not as well maintained as production servers, and are tempting targets. Organizations should ensure that DNS servers only store necessary records, and that all unauthenticated zone transfers are denied.

As an alternative to the dig command, websites such as ZoneEdit can be used to conduct DNS lookups [3]. These websites provide the same information as the dig command, but in a more attractive format.

In addition to performing zone transfers, the dig command can also be used to query DNS servers for their BIND version:

```
root@linux $ dig -t txt -c chaos VERSION.BIND @nameserver.company.com
;; <<>> DiG 9.1.0 <<>> -t txt -c chaos VERSION.BIND @ nameserver.company.com
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 32932
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0
;;
;; QUESTION SECTION:
;VERSION.BIND.                  CH      TXT
;; ANSWER SECTION:
VERSION.BIND.           0       CH      TXT     "8.1.2"
;;
;; Query time: 163 msec
;; SERVER: 10.1.1.1#53 (nameserver.company.com)
;; WHEN: Thu Feb 21 15:02:44 2002
;; MSG SIZE  rcvd: 60
```

In this case, the DNS server "nameserver.company.com" is running BIND 8.1.2 (see "8.1.2" in the dig output above). Equipped with this knowledge, attackers can search for specific exploits to BIND, which has historically been plagued by security holes.

- Traceroute

Like whois and dig, the "traceroute" command is built into the Linux operating system. Traceroute is extremely useful for mapping networks. By using incremental IP TTL (time-to-live) values and listening for ICMP time-exceeded messages, traceroute is able to display each intermediate system along the route from one host to another:

```
root@linux $ traceroute www.company.com
traceroute to www.company.com (10.1.1.2), 30 hops max, 38 byte packets
 1  192.168.1.1 (192.168.1.1)  1.659 ms  0.737 ms  0.696 ms
 2  atmrouter1.isp.net (192.168.2.1)  14.157 ms  13.769 ms  13.764 ms
 3  atmrouter2.isp.net (192.168.3.1)  14.686 ms  14.652 ms  14.477 ms
 4  borderrouter.company.com (10.3.1.1)  27.221 ms  25.697 ms  24.891 ms
 5  firewall.company.com (10.2.1.1)  27.010 ms  26.531 ms  26.253 ms
 6  www.company.com (10.1.1.2) 38.323 ms  64.957 ms  37.260 ms
```

Here a traceroute is performed to www.company.com (see the traceroute output above). The organization's border router and firewall are both

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

identified. It is surprising how many organizations use similarly descriptive names for important devices such as routers and firewalls.

In addition, by tracing the route to different servers, you can quickly discern the architecture of the organization's network, including the identification of internal routers and firewalls. To prevent this sort of information leakage, organizations should ensure that inbound ICMP is blocked by their border router or firewall. Some organizations may wish to allow some types of inbound ICMP to publicly accessible servers, such as corporate webservers. This is useful for the sake of reachability, and is generally acceptable.

- Websites such as the corporate website and the Edgar Database

  A wealth of useful information is available on the web. An organization's corporate website may have useful information such as contact names and phone numbers. Like the information learned by the whois command, this information can be used to launch social engineering and wardialing attacks, respectively.

  Also, some organizations will reveal the names of hardware and software partners on their websites. This information can be extremely useful. For example, if an organization is a partner of Check Point, you can bet they use FireWall-1. Whitepapers are an excellent source for both contact names and employed technologies. Whitepapers may also directly or indirectly reveal what security mechanisms are in place.

  In addition, a myriad of websites such as the Edgar Database have useful information. Specifically, the Edgar Database stores information pertaining to an organization's SEC filings, annual reports, and litigation [4]. Other websites offer valuable information regarding business news, financial data, and corporate assets.

- Usenet servers and associated websites

  Like websites, Usenet servers can also provide a wealth of useful information. For example, IT employees will sometimes post technical questions to newsgroups, asking how to configure a certain server, router, or firewall. These posts can provide invaluable technical details. Sometimes, the posts will actually reveal the presence of a vulnerability.

These tools can are all extremely useful during the Footprinting phase of a penetration test.

Scanning

The next phase of a penetration test is "Scanning". There are two distinct types of scanners – port scanners and vulnerability scanners. Port scanners probe hosts to identify open ports, while vulnerability scanners probe applications to identify known vulnerabilities. Both port scanners and vulnerability scanners are extremely useful tools.

An application scanner is a vulnerability scanner that scans a single application. A good example of an application scanner is a CGI scanner, which scans a

© SANS Institute 2000 - 2002, Author retains full rights.

webserver for known CGI script vulnerabilities. In addition, war dialers can also be considered scanners. War dialers dial a range of telephone numbers, searching for modems or PBXs. There are several useful tools that can be utilized during the Scanning phase:

- Nmap

  Nmap is the king of port scanners. Nmap is highly flexible, and can scan both TCP and UDP ports. In addition, Nmap supports several different scanning methods, performs ping sweeps, and provides OS detection. Nmap is run from the command line, and has several powerful options. Here the "-p0" option specifies not to ping the host before the scan, "-sS" specifies a stealth scan, "-p 1-1024" specifies to scan TCP ports 1-1024, "-O" specifies to detect the operating system, and "-v" specifies verbose output [5]. A stealth scan is used to evade detection. During a stealth scan, the three-way TCP handshake is intentionally not completed, causing most operating systems to not log the connection. The command performs the port scan and reports what ports are open and what operating system is being run:

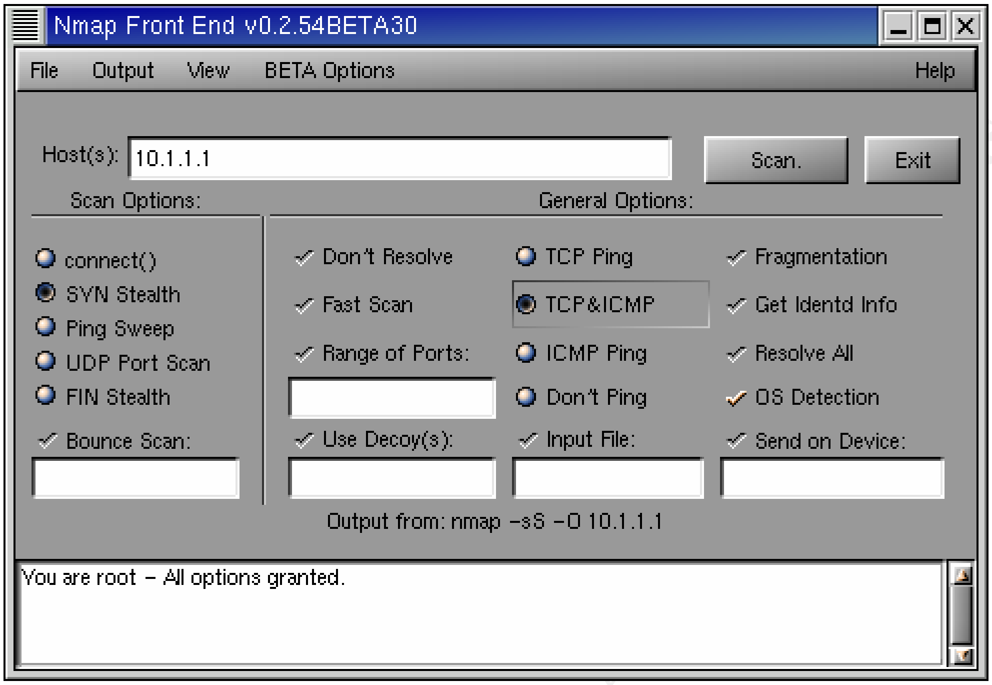
```
root@linux $ nmap -P0 -sS -p 1-1024 -O -v www.company.com
Starting nmap V. 2.54BETA30 ( www.insecure.org/nmap/ )
Host 10.1.1.1 appears to be up ... good.
Initiating SYN Stealth Scan against www.company.com (10.1.1.3)
Adding open port 80/tcp
Adding open port 443/tcp
The SYN Stealth Scan took 12 seconds to scan 1024 ports.
For OSScan assuming that port 80 is open and port 23 is closed
Interesting ports on www.company.com (10.1.1.3):
(The 1022 ports scanned but not shown below are in state: closed)

Port          State    Service
80/tcp        open     http
443/tcp       closed   https

Remote operating system guess: Linux 2.1.19 - 2.2.17
Uptime 24.278 days (since Sat Jan 26 09:01:57 2002)
TCP Sequence Prediction: Class=random positive increments
Difficulty=5173716 (Good luck!)
```

  In this case, the server www.company.com is running the HTTP and HTTPS services, and is most likely running the Linux operating system (see the Nmap output above). As you can see, Nmap is an incredibly useful tool. For penetration testing, Nmap is essential.

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

In addition, the NmapFE utility provides an intuitive GUI for Nmap [5]:



Host(s): 10.1.1.1

● SYN Stealth

● TCP&ICMP

✓ OS Detection

Output from: nmap -sS -O 10.1.1.1

```
You are root - All options granted.
```

As you select various Nmap options, the syntax of the Nmap command is dynamically generated. NmapFE is especially useful for new users, helping them learn the complex syntax of the Nmap command.

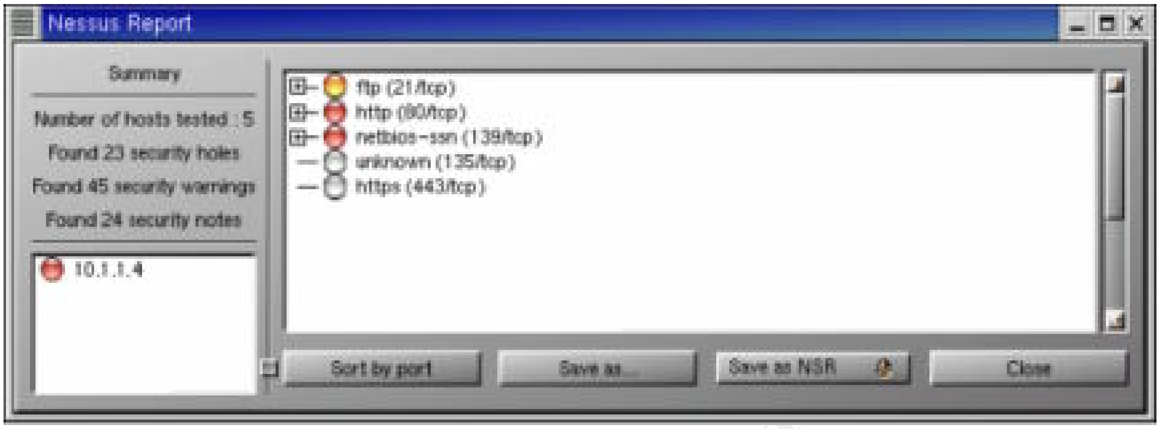
- Nessus

Nessus is an outstanding vulnerability scanner. The first popular vulnerability scanner was SATAN, the "Security Administrator's Tool for Analyzing Networks" [6]. Because of its devastating effectiveness, SATAN revolutionized the concept of hacking. Following in the footsteps of SATAN, Nessus is an open source project that "aims to provide to the internet community a free, powerful, up-to-date and easy to use remote security scanner" [7]. Nessus first scans for open ports, and then, based on available services, scans for the existence of known vulnerabilities. Nessus provides a detailed description of each vulnerability, and a corresponding link to the CVE website. CVE is the "Common Vulnerabilities & Exposures" project, and "aims to standardize the names for all publicly known vulnerabilities and security exposures" [8]. In other words, CVE provides a vendor-neutral vulnerability classification system.

Like Nmap, Nessus is highly flexible. Custom plugins can be created to test for specific vulnerabilities. And, since Nessus is an open source project, anyone can write plugins using NASL, the "Nessus Attack Scripting Language" [7].

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Nessus provides a simple GUI:


Nessus Report

Summary

Number of hosts tested : 5
Found 23 security holes
Found 45 security warnings
Found 24 security notes

10.1.1.4

ftp (21/tcp)
http (80/tcp)
netbios-ssn (139/tcp)
unknown (135/tcp)
https (443/tcp)

In this case, the server 10.1.1.4 is vulnerable to HTTP and Netbios-Session exploits. In addition, an FTP warning is found. Clicking on the "+" provides technical details of the vulnerabilities and warnings. As you can see, Nessus is an extremely powerful tool.

- SARA Top Twenty Vulnerabilities Scanner

  SARA (Security Auditor's Research Assistant) is a vulnerability scanner that was recently tweaked to scan for the vulnerabilities discussed in the SANS Twenty Most Critical Internet Security Vulnerabilities report [9]. This report allows users to "prioritize their efforts so they could close the most dangerous holes first" [10]. The tweaked version of SARA quickly and concisely scans for the existence of these twenty crucial vulnerabilities.

  Functionality wise, SARA is a vulnerability scanner very similar to Nessus. The tweaked version of SARA, however, allows administrators to concentrate on securing their systems against the twenty most exploited vulnerabilities on the Internet.

- Whisker

  Whisker is an optimized, flexible, customizable CGI scanner [11]. Whisker allows you to quickly scan a web server for devastating CGI script vulnerabilities. Whisker is written in Perl, and uses the Perl module "libwhisker" [11].

  Whisker is useful when a webserver employs CGI scripts. Instead of merely returning CGI scripts to the browser, webservers execute the contents of CGI scripts, and then return the results to the user. Some CGI scripts have known vulnerabilities. In other cases, if malicious user input is passed to the CGI script, it may not function as intended. Whisker scans a webserver's CGI scripts, actively looking for such anomalies.

- Shokdial

  Shokdial is a wardialer. Wardialers dial a list of telephone numbers, searching for modems or PBXs. Shokdial reads its configuration from and

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

writes its output to several different configuration files [12]. Although Shokdial does not provide a GUI, it is an efficient tool for wardialing.

Modems represent a backdoor into an organization's network. Even if an organization's firewall is secure, an unauthorized modem could compromise the entire network. For this reason, it is important to search for unauthorized modems on a regular basis.

These tools are all extremely effective during the Scanning phase of a penetration test.

Exploiting

The final phase of a penetration test is "Exploiting". During the Exploiting phase exploits are run against the target systems. There are several useful tools that can be utilized during the Exploiting phase:

- SecurityFocus

  The SecurityFocus Vulnerability Database is extremely useful for locating vulnerability exploits. The SecurityFocus Vulnerability Database allows attackers to quickly locate exploits by specifying a program's vendor, title, and version. A description of the vulnerability and the source code are provided [13].

  An exploit is code that will take actually take advantage of a vulnerability, yielding unauthorized access of some kind. After executing an exploit, the threat of the vulnerability has been realized. Websites such as SecurityFocus provide a centralized database of exploits.

- Search engines such as Google

  Another good place to search for exploits are search engines such as Google [14]. Exploits can be quickly located by searching for the word "exploit" along with the program's title and version. For example, searching for "exploit BIND 8.1.2" should return exploits for BIND 8.1.2.

- Mailing Lists such as Bugtraq

  Yet another good place to search for exploits are mailing lists such as Bugtraq. Bugtraq is a moderated mailing list used to discuss bugs in a variety of hardware, operating systems, and applications. Bugtraq is an excellent source for bleeding edge exploits that have not yet been posted to SecurityFocus or catalogued by search engines such as Google. Information regarding the Bugtraq mailing list, including instructions for subscribing, is available at SecurityFocus [15].

- Gcc & Perl

  The "gcc" command is built into the Linux operating system. Gcc is a compiler used to compile vulnerability exploits written in C. The "perl" command is an interpreter used to run vulnerability exploits written in Perl. Like gcc, perl may also be built into the Linux operating system. When you download exploits, they are usually comprised of source code written in C or

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Perl. To run the exploit, you either need to compile it using gcc or interpret it using Perl. For example, to compile the C language exploit contained in the file "exploit.c", you would use the following command:

```
root@linux $ gcc -o exploit exploit.c
```

Assuming the exploit compiles correctly, it can be run using the following command:

```
root@linux $ ./exploit
```

Before compiling an exploit, however, you should thoroughly review the source code. Some exploits are written with ulterior motives, so be careful. Know exactly what the source does *before* you compile it. Also, be sure that clients actually want you to run exploits against target systems. Some exploits may unknowingly crash target systems, so your mileage may vary. In addition, some exploits may require slight tweaking before they successfully compile. This prevents script kiddies from easily compiling and launching the exploits.

- John the Ripper

  John the Ripper is a password cracker used to "detect weak UNIX passwords" [16]. John the Ripper works by using dictionary files to generate password hashes. As each password hash is generated, it is compared to hashes found in the password file. If a match is found, that user's password has been cracked. In addition, John the Ripper performs obvious variations on dictionary words, such as appending numbers to words and substituting "3" for "e" [16].

  John the Ripper is useful when you are able to download a copy of the /etc/passwd file from a UNIX server. In other cases, a client may provide you with a copy of their /etc/passwd file so that you can audit the security of user passwords without having to independently obtain the password file. In either case, John the Ripper is usually able to successfully crack at least some of the passwords.

These tools are all extremely effective during the Exploiting phase of a penetration test.

<u>Other Miscellaneous Tools</u>

Although they do not neatly map to a specific phase of a penetration test, the following miscellaneous tools can also be very useful:

- Hping

  Hping is a reachability tool similar to "ping", but also supports TCP and UDP. In addition, hping supports "a traceroute mode, the ability to send files between a covered channel, and many other features" [17].

  Hping is an extremely useful tool for testing firewall rules. In addition, hping can be used to verify the existence of servers that cannot be pinged. For example, hping can be used to test connectivity to TCP port 80 on a web

© SANS Institute 2000 - 2002, Author retains full rights.

server. The “-S” option specifies to send SYN (synchronization) packets, and “-p 80” specifies port 80. The command reports host connectivity similar to the ping command:

```
root@kramer $ hping -S -p 80 www.company.com
HPING www.yahoo.com (eth0 10.1.1.2): S set, 40 headers + 0 data bytes
len=46 ip=10.1.1.2 flags=SA DF seq=0 ttl=46 id=26632 win=16384 rtt=51.4 ms
len=46 ip=10.1.1.2 flags=SA DF seq=1 ttl=46 id=28577 win=16384 rtt=51.2 ms
len=46 ip=10.1.1.2 flags=SA DF seq=2 ttl=46 id=30499 win=16384 rtt=51.1 ms
len=46 ip=10.1.1.2 flags=SA DF seq=3 ttl=46 id=32348 win=16384 rtt=51.2 ms

--- www.company.com hping statistic ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 51.1/51.2/51.4 ms
```

In this case, the server www.company.com responds with SYN|ACK (synchronization and acknowledgement) packets. In other words, www.company.com is running a webserver. In addition, by tweaking destination ports, hping can be extremely useful for testing firewall rules.

- Netcat

  Netcat is widely referred to as “the Swiss Army Knife of TCP/IP”. Netcat is a “simple Unix utility which reads and writes data across network connections” [18]. Netcat is useful for a variety of applications. For instance, netcat can be used for network performance testing, scanning, or can provide a backdoor shell into a system [18]. For example, suppose you have compromised a Linux server on a DMZ segment behind a firewall. Next, you want to launch a BIND exploit against another server on the DMZ. The problem is that the firewall does not allow inbound DNS access to the DMZ. The firewall does, however, allow inbound HTTP access to the DMZ. To get around this problem, netcat can be installed on the compromised Linux server and configured to redirect inbound HTTP requests to DNS requests on the target server. In the first netcat command the “-l” option specifies to listen, and “-p 80” specifies port 80. In the second netcat command (located after the “|”), the “10.1.1.4” and “53” arguments specify to redirect input to port 53 on the server 10.1.1.4. The command is run:

```
root@linux $ nc -l -p 80 | nc 10.1.1.4 53
```

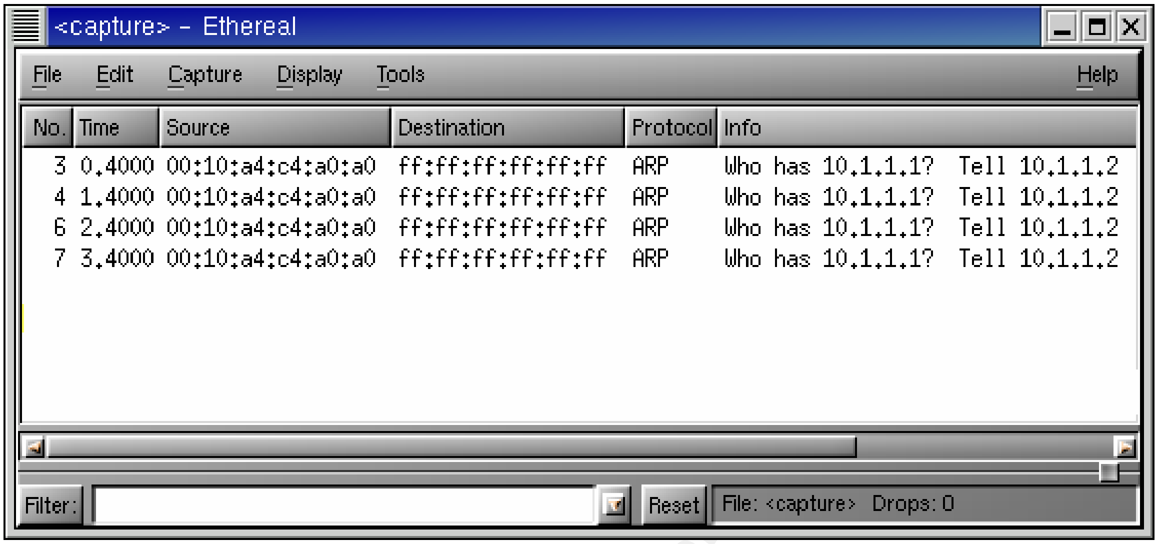
  In this fashion, netcat is able to redirect connections on the DMZ, effectively subverting the security of the firewall. Netcat has many other valuable applications. Because of its simplicity, netcat is an infinitely useful tool.

- Ethereal

  Ethereal is a superb network sniffer. A sniffer monitors a network and captures any packets that may be traversing the network. Ethereal is able to sniff switched networks, and provides protocol-specific decoding.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Ethereal provides an intuitive GUI:



```
No. Time   Source            Destination       Protocol Info
  3 0.4000 00:10:a4:c4:a0:a0 ff:ff:ff:ff:ff:ff ARP      Who has 10.1.1.1?  Tell 10.1.1.2
  4 1.4000 00:10:a4:c4:a0:a0 ff:ff:ff:ff:ff:ff ARP      Who has 10.1.1.1?  Tell 10.1.1.2
  6 2.4000 00:10:a4:c4:a0:a0 ff:ff:ff:ff:ff:ff ARP      Who has 10.1.1.1?  Tell 10.1.1.2
  7 3.4000 00:10:a4:c4:a0:a0 ff:ff:ff:ff:ff:ff ARP      Who has 10.1.1.1?  Tell 10.1.1.2
```

Here ARP requests for 10.1.1.1 from the server 10.1.1.2 are displayed. Ethereal supports filters, which control what types of traffic are captured. In addition, Ethereal has several built in protocol decoders that can be used to view protocol specific parameters [19]. Ethereal is useful for a variety of applications, such as sniffing passwords, network mapping, and network baselining.

These tools are all extremely effective during various phases of a penetration test.

Tool Summary

The following table provides a summary of all of the tools covered in this paper:

Tool	Description	Location
Whois	Footprinting tool	Built into Linux, or http://www.verisign-grs.com/cgi-bin/whois
Dig	DNS tool	Built into Linux, or http://www.zoneedit.com/lookup.html
Edgar Database	Information database	http://www.sec.gov/edgar.shtml
Nmap	Port scanner	http://www.insecure.org/nmap/
Nessus	Vulnerability scanner	http://www.nessus.org/
SARA	Vulnerability scanner	http://www.cisecurity.org/scanning_tool.html
Whisker	CGI Scanner	http://www.wiretrip.net/rfp/p/doc.asp/i2/d21.htm
ShokDial	War dialer	http://www.w00w00.org/files/ShokDial/
SecurityFocus	Vulnerability database	http://online.securityfocus.com/bid
Google	Search Engine	http://www.google.com/
Bugtraq	Mailing list	http://online.securityfocus.com/archive/1
John the Ripper	Password cracker	http://www.openwall.com/john/
Hping	Miscellaneous tool	http://www.hping.org/
Netcat	Miscellaneous tool	http://www.zoran.net/wm_resources/netcat_hobbit.asp
Ethereal	Network sniffer	http://www.ethereal.com/

It is important to note that this is by no means a comprehensive listing of Linux penetration testing tools. Rather, the tools covered in this paper are a starting point for a solid penetration-testing platform. Depending on the target of the penetration test, other tools will undoubtedly be needed.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

References

1. Red Hat Linux
   Multiple authors.
   Software available at http://www.redhat.com/products/.
   Documentation available at
   http://www.redhat.com/support/resources/install_upgrade/installing_linux.html.

2. VeriSign Global Registry Services Whois Page
   Available at http://www.verisign-grs.com/cgi-bin/whois.

3. ZoneEdit
   Available at http://www.zoneedit.com/lookup.html.

4. Edgar Database
   Available at http://www.sec.gov/edgar.shtml.

5. Nmap Port Scanner
   Written by Fyodor.
   Software and documentation available at http://www.insecure.org/nmap/.

6. SATAN Vulnerability Scanner
   Written by Dan Farmer and Wietse Venema.
   Software and documentation available at http://www.fish.com/satan/.

7. Nessus Vulnerability Scanner
   Multiple authors.
   Software and documentation available at http://www.nessus.org/.

8. CVE Website
   Available http://cve.mitre.org/.

9. SANS Top Twenty Vulnerabilities Scanner
   Written by Bob Todd.
   Software and documentation available at
   http://www.cisecurity.org/scanning_tool.html.

10. SANS Top Twenty Vulnerabilities
    Multiple authors.
    Report available at http://www.sans.org/top20.htm.

11. Whisker CGI Scanner.
    Written by Rain Forest Puppy.
    Software and documentation available at
    http://www.wiretrip.net/rfp/p/doc.asp/i2/d21.htm.

12. ShokDial Wardialer
    Written by Shok.
    Software and documentation available at
    http://www.w00w00.org/files/ShokDial/.

13. SecurityFocus Vulnerability Database
    Available at http://online.securityfocus.com/bid.

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

14. Google Search Engine
Available at http://www.google.com/.

15. Buqtrag Mailing List
Archive and mailing list information available at
http://online.securityfocus.com/archive/1.

16. John the Ripper Password Cracker
Written by Solar
Software and documentation available at http://www.openwall.com/john/.

17. Hping Miscellaneous Tool
Multiple authors.
Software and documentation available at http://www.hping.org/.

18. Netcat Miscellaneous Tool
Written by Hobbit.
Software and documentation available at
http://www.zoran.net/wm_resources/netcat_hobbit.asp.

19. Ethereal Network Sniffer
Multiple authors.
Software and documentation available at http://www.ethereal.com/.

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.