Global Information Assurance Certification Paper

The Art of Reconnaissance – Simple Techniques.

Sai Bhamidipati
Version 1.2e
August 18, 2001.

Prologue

Welcome to the world of Hacking!
After reading myriad articles on Internet security and hacking, i am convinced that every security conscious computer professional must learn the ways of the hacker. Reading is the only way out. Launching a cyber attack on a target is just like going into war. And like a good Commander who always makes intelligent guess of the enemy's strength and weakness we must first gain knowledge - **Reconnaissance** is the word for this. Cyber war is like a secret commando attack, wherein after the attack, no traces must be left. The attack and clearing traces is paper on its own and we shall only give some theory on it in the end. Let me warn you right here, that I have omitted a number of precautions taken by a experienced hacker, for instance they would never use their own phone number to break into any network or even reconnaissance. Reason is simple - Cyber attacks are stealth wars or gorilla wars, the opponent must never know from where you are attacking. For our discussion we will concentrate on basic and simple techniques using real world examples.

The paper is divided into 2 parts. In part I we will discuss Basic Reconnaissance and part II is all about Fingerprinting. The epilogue cites some articles on advanced techniques (You are better left to read the original works – its out of scope for current discussion)

Most of the commands we used in the following text were run from Redhat Linux 6.1.

Ok! Time to Begin

PART - I

In the text that follows we shall concentrate on reconnaissance with a motive i.e. trying to attack a particular target say victim organization. Victim organization is in India and all the information available with us is a domain name **victim.co.in** (as we go along we will confirm if this really belongs to the victim organization). With this knowledge how do we launch an attack against the victim organization?

Basic Reconnaissance

We need to collect more knowledge about the victim, so lets launch Operation Reconnaissance.

Step 1 – DNS

The DNS is our first link to gain knowledge on victim. From the ".in" at the end of domain name we know that victim.co.in is an Indian domain. Indian domains have a WHOIS client at http://domain.ncst.ernet.in/domreg/home.html. You can use the site

http://www.allwhois.com for querying most of the country level whois databases or try searching in google.com (which is what I did). The top-level domain information is hosted by the whois database of Internic. We can use the whois client to query this database as **root # whois xyz.com.**
The whois client gave us the following information against victim.com

Domain Name: victim.co.in
Address of victim organization
Administrative and technical contact for victim.co.in
Primary and Secondary NameServer's Name and IP Address.

The Address of victim organization confirms the ownership of our domain victim.co.in. The primary and secondary name servers are the authoritative servers for victim.co.in. The administrative and technical contacts can be used for Social Engineering attack as a last ditch effort, if none of our techniques lead us to success.

Any idea where we are heading now?

Step 2 - Getting IP Addresses

We need to know if victim organization has a network with public IP's of which victim.co.in is a part. This leads us to the knowledge of IP number allocation. IP numbers are divided among Asia Pacific, Europe and Africa, and North and South America. The organizations responsible for IP delegation are APNIC, RIPE and ARIN respectively.

Since India is in Asia we look towards APNIC and again our friend whois is at our rescue.
We have two ways to go now. One, run our own whois which might take some learning curve on using whois, and two, to use some site which supports a whois client for complete Internet. I chose option two (as the title says – simple techniques). The site to use is http://www.samspade.org/. There a number of online tools and the one we used is the whois online tool from Sam Spade's site.

Wow! The whois actually returned along with other information - the IP's allocated to victim organization. APNIC has allocated 64 IPS belonging to a class C network from x.y.z.0 - 63 to our victim (I apologize but I have to sanitize all information and you shall only get to see a lot of x.y.z's *grin*). We can easily calculate that victim uses a net mask of 255.255.255.192.

Getting a fair idea about this reconnaissance thing, are you?

Step -3 Tracing the IP

How far is the victim IP from you? This information will tell us one of the routes between victim's IP and us. Traceroute will do this trick for us.

root # traceroute victim.co.in

Check the second last IP. This is usually a packet filtering router or firewall. In our result we find that the second last IP is a different one from those allocated to victim's

organization. Most probably it is a victim end router provided by the ISP through which victim's network connects to the Internet.

Step 4 - DNS revisited with host command

root # host -l -v -t any victim.co.in
Truncated output of above command -
rcode (0)
Found 1 address on ns.xyz.co.in
Found 1 address on ns2.xyz.co.in

We find that these are the same name servers as those shown in whois database. Host command also gives us the fully qualified domain name of the mail exchange server given a domain.

root # host victim.co.in
Output of above command -
victim.co.in has address x.y.z.2
victim.co.in mail is handled (pri=5) by mail.victim.co.in
A further ping on mail.victim.co.in reveals that its IP Address is x.y.z.1

Now, wouldn't it be nice to attach Operating Systems and Services running, to each machine on the victim's network? – The answer to this is, our discussion in part ii.

Part - II

A. Ping Sweeps, Fingerprinting and Port Scanning.

How do we ever know if a machine is up, what services its running etc? Let me feed you a brief introduction.

A simple ICMP ECHO request can be sent to a remote machine and if its up it will reply with a ECHO response, else you will get a host unreachable message. Simple, but how to do this? The ping program supplied with most operating systems does this for you. Ping will only scan one IP for you at a time and thus can't be used for simultaneous ping sweeps. For scanning a bunch of IP's what you need is a tool like hping written by a person under the alias antirez, or maybe fping. Pinging many IP addresses simultaneously is known as ping sweeps. In some cases ICMP packets are blocked by the firewalls, so what you would like to do is scan for port 80, which most firewalls let go through if the network hosts a web server – This is called TCP scanning.

The method for detecting the remote operating system is fingerprinting. Basic fingerprinting is done by either simple ftp, telnet to port 23, 25 or 80 and collecting banner information. Most banners display the operating system name in their default configuration. Advanced fingerprinting analyses the TCP stack implementation. A TCP packet is crafted by switching ON or OFF certain flags and sent to the remote machine. The remote operating system based on its TCP stack implementation sends a response, with some

specific flags ON or OFF (most often used flags are the SYN, ACK and FIN flags). Depending on TCP responses collected for each crafted packet we can make an intelligent guess of the operating system from its database of TCP stack signatures. The latest copy of Operating System fingerprints can be found at http://www.insecure.org/nmap/nmap-os-fingerprints.

Portscanning involves opening a normal connection to each TCP port on remote machine and sending a TCP packet with SYN flag set – The first step in a TCP three way handshake for initiating a TCP dialogue. An open port always replies with a TCP packet with the SYN and ACK flags set. A closed port sends a packet with RST and ACK flags set. There are a number of stealth scans to avoid detection which usually never complete the connection they initiate or follow a proper connect sequence, for instance the TCP half open scans where the attacker tool sends a SYN packet. On receiving a SYN + ACK packet from the server it immediately sends a RST packet and closed the connection.

For the purpose of illustration we will use just one tool called **nmap** (by a person with a alias fyodor) which easily accomplishes OS detection, ping sweeps and port scanning – **A all in one tool**.

Note - The 'Download Links' at the end of this paper has links to additional tools.

The commands used are not very neat and should be tweaked to suit your needs. And the best way to do that is by reading the nmap man page.

root # nmap -sS -O -I -v -oN /tmp/portscan_results -p 20-30,53,69,110,111, 139, 140, 3128, 4166, 6000-6063, 6666 x.y.z.1-63

In the above example we are trying all well-known ports at once, this is not a good idea and is used only to expedite our work. Normally if we have an exploit for a particular service and we know the port it runs on we would just run nmap against that port.

Nmap when done saves the results in human readable format to the file you specify using the –oN option. In above example the results are saved to the file /tmp/portscan_results.

Opening that file we see some interesting information. The information is sanitized as well as edited, for instance the TCP Sequence number guessing difficulty information is removed since it is not relevant to our discussion.

Interesting ports on (x.y.z.1)

Port	State	Service	Owner
21/tcp	open	ftp	
22/tcp	open	ssh	
25/tcp	open	smtp	
53/tcp	open	domain	
80/tcp	open	http	
110/tcp	open	pop-3	

Remote operating system guess: Linux 2.2.12

Interesting ports on (x.y.z.10):

(The 90 ports scanned but not shown below are in state: closed)

Port	State	Service	Owner
23/tcp	open	telnet	
79/tcp	open	finger	

Remote OS guesses: AS5200, Cisco 2501/5260/5300 terminal server IOS 11.3.6(T1), Cisco IOS 11.3 - 12.0(9)

The surprising result is x.y.z.10, since it is running a router here (The Cisco 2501/.. part in above Remote OS guess). Let's run a finger this IP using the finger client for more information. For the finger client to work the remote machine much run a finger service and from above nmap result we know that the finger server is running on port 79.

root # finger @x.y.z.10
Truncated output below -
[x.y.z.10]

Line	User	Host(s)	Idle	Location
226 vty 0		*.*.*.*	12:19:00	*.*.*.*
*227 vty 1		idle	00:00:00	*.*.*.*

Interface	User	Mode	Idle	Peer Address
Se1/2		Sync PPP	00:00:00	*.*.*.*

The *'s above are used to sanitize actual IP's but from above data I am convinced that it is router accepting dial-up connections. Let do some further scanning to verify this assumption.

root # nmap -sU -p 1812, 1813 x.y.z.1-10

This tells us that x.y.z.1 and x.y.z.2 are running radacct service on udp port 1813 and x.y.z.10 is running radius service on udp port 1812. RADIUS is the remote authentication dial-in user service and radacct is the server that does accounting for each dial-in user. So our assumption is verified. So victim organization might be a small ISP.

B. Services – One more passage to reconnaissance

We know the services running on say x.y.z.1. The question to ask now is – how will this information help us. Above we discussed some advanced techniques using nmap, now I'll walk you through some basic ways for fingerprinting.

1.Let start with ftp
root # ftp x.y.z.1
The banner information gives out that the ftp software is **Version wu-2.6.1(1).**
2. SSH client connect to port 22 let out the software and version of SSH server to be SSH-1.99-OpenSSH_2.2.0pl. We cannot telnet to a SSH server. So we download a SSH client and run it.

3. Telnet to SMTP port 25 will tell us something about mail server software running.

root # telnet x.y.z.1 25

Unfortunately the **vrfy** or **expn** commands were not supported. When I sent the command help to the server I got the following reply
"**qmail home page: http://pobox.com/~djb/qmail.html**". From this I guessed the SMTP server is **Qmail**.

Version I couldn't figure out. Qmail is supposed to be very secure so we leave this alone for now. But if you know any exploits for it just incase – this turns out to be very important information.

4. A telnet to port 80 revealed the most interesting information.

root # telnet x.y.z.1 80
On the prompt I typed in the following **HTTP/1.1 /POST /cgi-bin/xyz**
Note: the /POST was a typo instead of just 'POST' but gave a list of acceptable methods too so I retained it here.
Web Server output below -

Server: Apache-AdvancedExtranetServer/1.3.14 (Linux-Mandrake/2mdk) PHP/4.3.3pl1
ApacheJServ/1.1.2
Methods: GET, HEAD, OPTIONS, TRACE

Now, now, what more can we ask. This output has narrowed down even the distribution of Linux running, which is Linux-Mandrake/2mdk and also has given us a plethora of information.

Now you are at a stage where every intelligent Commander is. Ready with information about all enemy weaknesses and thus better prepared to Attack. But Gee I apologize, as my paper ends here – maybe some other day I will take your hand and guide you through an Intelligent attack, but before that remember what I said – Read, Read and Read. No other way in this field of Information Security and Information Theft.

Epilogue

If you have heard of packet filtering and logging tools, firewalls and Intrusion Detection Systems (IDS) then the above discussion might immediately switch a bulb – Will these techniques put me a bad spot against those tools. The answer tragically is 'Yes'. I have used the above techniques only to demonstrate some simple techniques that don't take these tools into consideration.

How can better the above techniques to safeguard you against these tools? I will answer this in short in the discussion that follows. These techniques need hands-on expertise and are highly complicated. I couldn't get much from any guides or HOWTO's I read on the Internet.

- Never connect to the Internet using your own telephone line. Learn more phreaking from any resource you may find. Understand the telecom technology used in your area, including simple things like how many wires, what colors and where these wires

connect and to which gadgets. For starters follow the telephone line from your house and see where and how it connects to the telephone jack, the local junction etc.

- The above rule is not so simple always. So what other things can you do? Spoofing IP's is the next best thing. Spoofing is a way we can send packets to a remote machine using someone else IP. Ensuring that the IP you spoof is up and is an idle host helps – A very interesting article on this is by Chmielarski, Tom. "Reconnaissance Techniques using Spoofed IP Addresses" April 4, 2001. URL - http://www.sans.org/newlook/resources/IDFAQ/spoofed_IP.htm.
- Don't use the nmap command as outlined in this paper. Use nmap with the **decoy option –D** to make your scans less detectable**.** This option allows you to specify a number of IP addresses and you can hide your own IP among these using **ME**. The nmap man page states that one must place the ME as the sixth IP in your decoy list. One more important thing to remember is that we are using the decoy option to trick the IDS, so don't use IP's which don't exist or which are not up at that time for decoy purpose. Otherwise the task of IDS is made very simple, as other hosts will not send any packets in reply.

A FINAL NOTE – You have seen a lot of action here. You may want to try all this out on your own. Let me tell you one thing. The above techniques were run against a friendly network that gave me the go ahead. By friendly I mean it belongs to my best buddy, and I promised to secure his entire network against this favor. It was a go ahead by word of mouth, but I trust him. What I would suggest is before planning to increase you knowledge, setup a small cheap network of your own. If you can get such a friend like I did, please do take written permission. Never and I repeat never run any reconnaissance techniques in the wild, since it will only end up getting you in trouble with the law. Rest is upto you.
That's all for now folks. If this seemed boring, I hope you will enjoy the articles in the reference section.

References

1. Antirez. Bugtraq post: "new tcp scan method", December 18, 1998. URL - http://www.kyuzz.org/antirez/papers/dumbscan.html.
2. Fobic. "Examining Advanced Remote OS Detection Methods/Concepts using Perl", Feb 03, 2001 URL - http://www.packetnexus.com/kb/greyarts/docs/981766898:16776.html.
3. Fyodor. "Nmap network security scanner man page", URL - http://www.insecure.org/nmap/nmap_manpage.html.
4. Lamont Granquist. Email to nmap-hackers@insecure.org: "NMAP guide", Mon, 5 Apr 1999 URL - http://www.insecure.org/nmap/lamont-nmap-guide.txt
5. Fyodor. "The Art of Port Scanning", URL - http://www.insecure.org/nmap/nmap_doc.html
6. dethy@synnergy.net. "Examining port scan methods - Analysing Audible Techniques", (C)opyright 2001 by dethy@synnergy.net. URL - http://secinf.net/info/misc/portscan.html
7. Kamerling, J. Eric. "The Hping2 Idle Host Scan", February 26, 2001 URL - http://www.sans.org/infosecFAQ/audit/hping2.htm.

Bibliography

1. Comer, E. Douglas. "Internetworking with TCP/IP Vol. I: Principles, Protocols, and Architecture" 3rd Edition Volume 001. Prentice Hall, March 24 1995.
2. Albitz, Paul & Liu, Cricket. "DNS and BIND" 4th Edition. O'Reilly & Associates, Inc. April 2001.
3. McClure, Stuart & Scambray, Joel & Kurtz, George. "Hacking Exposed: Network Security Secrets and Solutions" McGraw Hill Professional Publishing, Sept 10, 1999.

Download Links

1. Nmap - www.insecure.org/nmap/. On this page you will find the download link to the latest nmap tool.
2. Hping - http://www.hping.org/download.html. You will get the latest hping source here.