Many companies only concentrate on protecting their systems from a specific exploit when they start building a security infrastructure. They figure out what patches need to be applied to their systems, and after they apply them, they think they are secure. However, they do not realize that through reconnaissance and information gathering, an attacker can acquire a large amount of information about their sites.

Before an attacker can run an exploit, he needs to understand the environment he is going after. In doing so, he needs to gather preliminary information about the number of machines, type of machines, operating systems, and so forth. If someone was going to rob a bank, they would not just wake up one day and randomly pick a target. They would scope out the possible targets and gather information about how the bank works, where the guards stand, when they change shifts, possible weaknesses that can be exploited, and based on that information, they would decide not only which target to attack, but how to attack it. No matter what the target is, before an attacker goes after it, he has to gather as much information as possible, so his chances of success are very high. In most cases, whether an attack is successful or not is directly related to how much information was gathered about the target. As you will see, if an attacker performs the information gathering stage correctly and in enough detail, access is almost guaranteed.

Therefore, it is key for a company to know what information an attacker can acquire about it and minimize the potential damage. When I perform security assessments, I perform information gathering against a company to try to find out its points of vulnerability. In doing so, I acquire a lot of useful information about the site. In some cases, I take the information and produce a network map of the company, and in several cases, the end result was a better map than the companie's IT department had. The question I pose is this: After an attacker has a detailed map of your network and knows exactly what software and versions are running on each machine, how hard is it for him to successfully exploit your network? The answer is simple. After someone has that much information, the network is as good as compromised. Therefore, it is key that an attacker only gains limited information about a network

Steps for Gathering Information

The following are the seven basic steps an attacker would take to gather information about a target. After each step are some of the tools an attacker would use to gain the information he needs to exploit the target:

1. Find out initial information:
    - Open Source
    - Whois
    - Nslookup
2. Find out address range of the network:
    - ARIN (American registry for internet numbers)
    - Traceroute
3. Find active machines:
    - Ping
4. Find open ports or access points:
    - Portscanners:
    - Nmap
    - ScanPort
    - War Dialers
    - THC-Scan
5. Figure out the operating systems:
    - Queso
    - Nmap

6. Figure out which services are running on each port:
    - Default port and OS
    - Telnet
    - Vulnerability scanners
7. Map out the network:
    - Traceroute
    - Visual ping
    - Cheops

In this chapter, we will take a look at each of the seven steps and examine how each of the tools work. Not only will we see how they can be used by an attacker to compromise a system, but we will show you how to use them to protect your system. Most people have a negative view towards tools that can be used to compromise systems because they fail to realize the benefit of using these tools. If you understand and use these tools on a regular basis, they can be used to increase the security of your site. Also, if you use them to increase your security and protect your site, then the value of these tools to an attacker decreases. The thing to learn from this chapter is that these tools should be embraced. The more you know and understand how an attacker breaks into a network helps you increase the security at your site. After we cover all the steps and tools, we will finish the chapter with an example of *red teaming*, which shows how you can simulate an attack to determine and fix your vulnerabilities before a real attacker exploits them.

Find Out Initial Information

For an attacker to compromise a machine, he needs to have some initial information, such as an IP address or a domain name. In this chapter, we will be assuming that the system the attacker is targeting uses a static IP address, which is true for most servers. A static IP address is where the IP address stays the same each time the system is rebooted. This is the opposite of a dynamically assigned IP address, which could potentially change each time the system is rebooted.

If an attacker is specifically going after your site, he will know your address well in advance and will concentrate solely on compromising your network. This occurs in situations where a company has information that an attacker wants. For example, in cases of corporate espionage, a company wants my company's trade secrets, and therefore, will target my company's network and no one else's.

Other attackers just randomly scan the Internet looking for networks that either look easy to compromise or look like they might have valuable information. For example, attackers who run distributed attacks against other companies need machines they can use to launch the attacks. They do not care whose machines they are, as long as they can be

compromised in a short period of time. This is why it is so important that you tighten your security as much as possible. If someone does a basic port scan and does not find a lot of open ports, but he scanned twenty other networks that have open ports, he might pass up your network and go after someone else's. The key is not to look like an attractive target.

Now that an attack has a given domain name, the attacker will need to gather information about the site. Information, such as IP addresses or people who work at the site, can all be used to help launch a successful attack. Now let's look at some ways that can be used to gather the initial information.

Open Source Information

In some cases, companies give away large amounts of information without knowing it. Information that a company thinks is general information or information that could help bring in clients could also provide useful information that would greatly simplify an attacker's job. This information is generally called *open source* information. Open source is general information about a company or its partners that anyone can obtain. This means that accessing or analyzing this information requires no criminal element and is perfectly legal. Because of this, it is key for companies to control and limit the information they give away.

Let's look at an example.A company that provides managed services just built a state of the art network operations center, and in the goal of attracting customers, it posts a press release to its web site. The press release states something like the following:

"Company X is proud to announce the opening of its new state of the art network operations center. Company X has built a premier center to provide its customers with the best monitoring capabilities around. All monitoring stations are running Windows 2000, which access data across a state of the art Cisco network consisting of Cisco's latest switches and routers. In addition, HP Openview and Nervecenter are used to monitor the systems with several Solaris workstation and servers. The center also has three points of access to Internet, which provide a high level of fault tolerance."

As you can see, this might attract new clients, but it is also giving attackers a road map for how to compromise the network. For example, if an attacker is going to launch a Denial of Service attack against this company, the attacker knows he has to take down three points of connectivity to the Internet. This information will help an attacker do his homework before an attack, which increases the success of his attack. Telling customers that their operations center has been upgraded with state of the art equipment is one thing, but giving specifics is probably

giving away too much of the farm. If a customer is really interested, let them call a sales person to tell them about all the great equipment, but do not give it away to the public. Not only does this give an attacker valuable information, but a company is also showing its cards to a competitor, which can use this information against them.

Support staff at a company need to know what information a company is giving away. For example, on most web sites companies have a directory listing where someone can find out not only the CEO and COO, but possibly who the VPs and directors of the company are. If the help desk does not know this information is publicly available and an attacker calls up claiming he works for an individual, the help desk might view this as sufficient information to believe the attacker. Therefore, it is key for companies to limit the individuals listed on a public site, and in cases where they want to showcase a member of the executive team, they should make sure everyone is aware that this information is publicly available.

Not only would an attacker search a company's web site, but he might search related web sites. For example, publicly-traded companies have to register and provide information to the government, and the government makes these databases available through the Internet. One such database is run by the SEC and is called edgar: http://www.sec.gov/edgarhp.htm. News groups also provide valuable information. One reason is that more and more support staff are using news groups to help solve their company's problems. For example, if a company is having problems with a mail server, an IT person might post a question to a news group asking for help. Looking at these requests and matching them to the company name in the email can provide a lot of useful information. Not only does it tell an attacker what equipment a company has, but it helps him gauge the sophistication of the staff. Also, partners like to link to each others' sites to help drum up business. The Altavista search engine has a feature called link. Typing `link:` followed by a URL address in the search field will tell you every site that has a link to the URL referenced. This can quickly identify a company's partners. You also might be quite surprised by who is referencing a company's site. In one case, an underground site actually had links to all the sites it has compromised. If a company is on that list, it might want to know about it.

Finally, what is available to a casual browser and what files actually exist on a company's web server are two different things. When an attacker connects to a web site, he clicks links to navigate the site, and by doing this, he can only access pages that are directly accessible to the links. A lot of sites also have what they call orphan pages, which are pages that exist on the web server but are not directly accessible because they are not linked by any page. These orphan pages can be accessed if the attacker knows the name of the file. An easier way is to use one of the

many web spider programs to download an entire site. This will give the attacker a list of every page that is on the server. This usually provides valuable information because web developers upload test pages, but never remove them, and because they are not directly linked to any other page, the developer thinks they are safe. I have done this and downloaded sample pages that contained active accounts and other useful information.

A company can never remove all open source information, however by being aware of it, the company can do things to minimize the potential damage. As you will see with whois, any company that has a domain name must give away certain information.

Whois

To gather information, we need an address or a starting point. With the Internet, the initial address usually takes the form of a domain name. For our examples, the attacker is going to use the domain name of newriders.com, although some of the information has been changed to protect the innocent. The first thing an attacker is going to do is run the whois program against this domain name to find out additional information. Most versions of UNIX come with whois built in. So, the attacker could just go to a terminal window or the command prompt and type whois newriders.com. For help, the attacker could type `whois ?` to get a listing of the various options. The following are some of the options available with whois 1.1 for Linux:

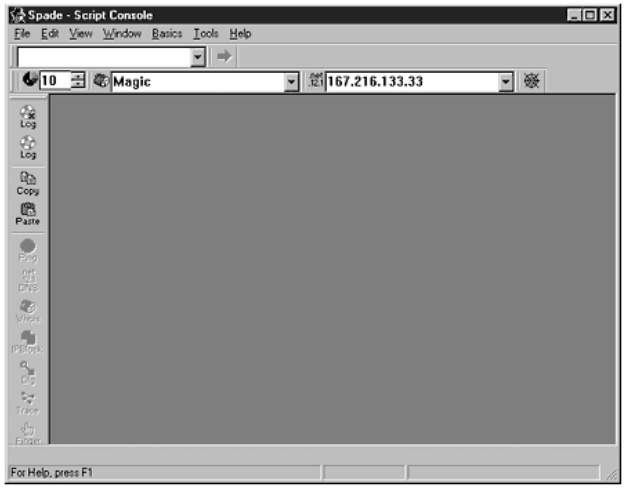
```
Whois Server Version 1.1

Domain names in the .com, .net, and .org domains can now be
registered with many
different competing registrars. Go to http://www.internic.net
for detailed
information.

Enter a a domain, nameserver, or registrar to search for its
information. You may
also search for nameservers using IP addresses.  WHOIS will
perform a broad search
on your input.  Use the following keywords/characters to
narrow your search or
change the behavior of WHOIS.

  To search for a specific record TYPE:
  -------------------------------------
   domain
   nameserver
   registrar
```

```
  Other WHOIS keywords:

    Expand                    Show all parts of display without
asking.
    FUll or '='               Show detailed display for EACH
match.
    SUMmary or '$'            Always show summary, even for
only one match.
    HELP                      Enters help program for full
documentation.
    PArtial or trailing '.'   Match targets STARTING with given
string.
    Q, QUIT, or hit RETURN    Exits WHOIS.

Your search will match everything BEGINNING with your input if
you use a trailing
period ('.') or the 'PArtial' keyword. For example, entering
"domain mack." will
find names "Mack", "Mackall", "MacKay".  The "domain",
"registrar", and
"nameserver" keywords are used to limit searches to a specific
record type.

     EXAMPLES:
     domain root
     nameserver nic
     nameserver 198.41.0.250
     registrar Network Solutions Inc.
     net.
   = net
     FU net
     full net
     $ ibm.com
     SUM ibm.com
     summary ibm.com


Search for a domain, nameserver, or registrar using its full
name to ensure that a
search matches a single record. Type "HELP" for more complete
help; hit RETURN to
exit.
>>> Last update of whois database: Wed, 19 Jul 00 03:09:21 EDT
<<<

The Registry database contains ONLY .COM, .NET, .ORG, .EDU
domains and Registrars.
```

With Windows operating systems, the attacker would have to get a third-party tool to perform whois lookups. There are several available on the

Internet with different features and prices. A good starting point is to go to http://www.tucows.com, search whois, and get a long list of various programs that perform whois queries. The one I prefer is called Sam Spade and is also available at tucows. When you start up Spade, you get the screen shown in Figure 3.1.

Figure 3.1. Initial screen of Sam Spade.



Spade has a lot of utilities, not just whois, so it is a handy tool to have. Most of the steps we talk about in this chapter can be accomplished with Spade. We will talk about other tools, because in some cases, they are a little more straightforward or provide additional information.

Now that an attacker has the tools he needs, he would run a whois query on the targeted domain, newriders.com, and obtain the following information:

```
whois newriders.com is a domain of USA & International
Commercial
Searches for .com can be run at http://www.crsnic.net/

whois -h whois.crsnic.net seccomputing.com ...
Redirecting to NETWORK SOLUTIONS, INC.

whois -h whois.networksolutions.com seccomputing.com ...

Registrant:
Eric C (NEWRIDERS-DOM)
```

```
   12345 Some Drive
   Somewhere, SA 20058
   US

   Domain Name: NEWRIDERS.COM

   Administrative Contact, Technical Contact, Zone Contact,
Billing Contact:
      C, Eric  (EC2515)  ERIC@someaddress.COM
      Eric C
   12345 Some Drive
   Somewhere, SA 20058
   US
      (555) 555-5555 (FAX) (555)555-5555

   Record last updated on 22-Jul-1999.
   Record expires on 17-Apr-2001.
   Record created on 17-Apr-1998.
   Database last updated on 19-Jul-2000 04:37:44 EDT.

   Domain servers in listed order:

   MAIL2.SOMESERVER 151.196.0.38
   MAIL1.SOMESERVER 199.45.32.38
```

By looking at this output, an attacker would get some very useful information. First, he gets a physical address, and some people's names and phone numbers. This information can be extremely helpful if an attacker is launching a social engineering attack against your site. An attacker basically has general information about the company and names and phone numbers for key people in the organization. If an attacker calls up the help desk and inserts this information into the conversation, he could convince the help desk that he does work for the company, and this can be used to acquire access. Because the people listed in the whois record are usually pretty high up and well known in a company, most people will not question the information that is being requested. So, if an attacker calls up and says, "I just got put on this sensitive project and Eric C told me to call up and get an account immediately, and I have his number if you would like to call him". Most technical staff would not realize that someone could get this information from the web, so they would think the request was legitimate and would probably process it.

Going to the end of the whois listing, we have two very important IP addresses, the primary and secondary name servers that are authoritative for that domain. An attacker's initial goal is to get some IP addresses of machines on the target network, so he knows what to attack. Remember, domain names are used because they are easier for humans to remember, but they are not actually addresses for machines. Every machine has to have a unique address, but it does not have to have a unique domain

name. Therefore, the unique address that an attacker is looking for is the IP address. The more IP addresses an attacker can identify as being on the target's network, the better chance he has of getting into the network.

Nslookup

One way of finding out additional IP addresses is to query the authoritative *domain name servers* (DNS) for a particular domain. These DNS servers contain all the information on a particular domain and all the data needed to communicate with the network. One piece of information that any network needs, if it is going to send or receive mail, is the MX record. This record contains the IP address of the mail server. Most companies also list web servers and other IPs in its DNS record. Most UNIX and NT systems come with an nslookup client built in or an attacker can use a third-party tool, such as Spade.

The following is the output from running nslookup:

```
03/28/00 12:35:57 dns newriders.com
Mail for newriders.com is handled by server1.newriders.org
Canonical name: new riders.org
Addresses:
 10.10.10.5
 10.10.10.15
```

Now an attacker has a couple of IP addresses that are on the domain. This can be used to start mapping out the network.

Another simple way to get an address is to ping the domain name. In cases where an attacker only has a domain name, he can either perform a reverse lookup or he can just ping the domain name. When trying to ping a domain name, the first thing the program does is try to resolve the host to an IP address, and it prints the address to the screen. The following is the output from the `ping` command:

```
Pinging newriders.com [10.10.10.8] with 32 bytes of data::
Request timed out.
Request timed out.
Ping statistics for 10.10.10.10:
Packets: Sent = 2, Received = 0, Lost = 2 (100% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum =  0ms, Average =  0ms Control-C
```

Now an attacker has a couple of addresses on the network that can be used as a staring point. It is important to note that I am using the 10.x.x.x addresses in my examples just to make sure we do not upset a

company by using its legitimate IP addresses. The 10 network is a private, non-routable address and, therefore, should be fairly safe to use.

One other note is that if a company is using a virtual ISP to host its web site, an attacker could receive various addresses when he performs an nslookup. A virtual ISP is where a single server is actually hosting several sites for various companies. It is important to realize this and be able to filter out which are the company's IP addresses and which are someone else's. The easier way to figure this out, in most cases, is the mail will go directly to the company. So, if the mail and web addresses differ significantly, an attacker might want to do a reverse lookup on the IP addresses of the web servers. If they belong to an ISP, then those addresses are outside the range of the company and should be ignored.

Find the Address Range of the Network

Now that an attacker has the IP addresses of a couple of machines, he wants to find out the network range or the subnet mask for the network. For example, with the address 10.10.10.5, without knowing the subnet mask, the attacker has no way of knowing the range of the address. The main reason he wants to know the address range is to make sure he concentrates his efforts against one network and does not break into several networks. This is done for two reasons. First, trying to scan an entire class A address could take a while. Why would an attacker want to waste his time, if the target he is going after only has a small subset of the addresses? Second, some companies have better security than others. Going after a larger address space increases the risk because now an attacker might break into a company that has proper security, and that company would report the attack and set off an alarm. For example, if the subnet mask is 255.0.0.0, then the entire 10 network belongs to that company, and an attacker can go after any machine. On the other hand, if the subnet mask is 255.255.255.0, then he can only go after 10.10.10.x because 10.10.11.x belongs to someone else.

An IP address is actually composed of two pieces: a network portion and a host portion. All computers connected to the same network must have the same network portion of the address but different host addresses. This is similar to houses. Two houses on the same block must have the same street address but different house numbers. The subnet mask is used to tell a system which part of the IP address is the network portion and which part is the host portion. For more information on IP addresses and subnets, see "*TCP/IP Illustrated, Volume 1*", by Richard Stevens.

An attacker can find out this information two ways, an easy way and a hard way. The easy way is to use the *American Registry for Internet Numbers* (ARIN) whois search to find out the information. The hard way is to use traceroute to parse through the results.

ARIN

ARIN lets anyone search the whois database to "locate information on networks, autonomous system numbers (ASNs), network-related handles, and other related Points of Contact (POCs)." Basically, the normal whois will give someone information on the domain name. ARIN whois lets you query the IP address to help find information on the strategy used for subnet addressing and how the network segments are divided up. The following is the information an attacker would get when he puts in our IP address of 10.10.10.5:

```
Some Communications (NET-SOME-ICON3) SOME-ICON3
                          10.10.0.0 - 10.10.255.255
NewRiders  (SOME-NewRiders)    ICON-NET-BA-NEWRIDERS
                          10.10.10.0-10.10.10.255
```

In this case, an attacker can see that New Riders acquired its IP addresses from Some communications, and Some communications has the range 10.10.x.x, which it subnets to its clients. In this case, New Riders was given the range 10.10.10.x, which means it has 254 possible hosts from 10.10.10.1 to 10.10.10.254 (remember host addresses of all 1's or 0's is invalid, so .0 and .255 cannot be used for a host address). Now an attacker can concentrate his efforts on the 254 addresses as opposed to the entire 10 network, which would take a lot more effort.

ARIN whois has a lot of different options that can be run. The following are some of the different options with examples, taken from http://www.arin.net.

```
Output from ARIN Whois

ARIN's Whois service provides a mechanism for finding contact
information for
those who have registered "objects" with ARIN. ARIN's database
contains Internet
network information including ASNs, hosts, related POCs, and
network numbers.

ARIN's Whois will NOT locate domain related information or
information relating to
Military Networks.  Please use rs.internic.net to locate
domain information and
nic.mil for NIPRNET information.

To locate records in our database, you may conduct a web based
Whois search by
```

inserting a search string containing certain keywords and characters (shown below
with their minimum abbreviation in all CAPS).

You may search by name, ARIN-handle, hostname, or network number.
Your results will be more or less specific depending on the refinements you apply
in your search. Follow the guidelines below to make your search more specific and
improve your results.

Using a Local Client

UNIX computers have a native whois command. The format is:

Whois -h hostname identifier e.g. Whois -h rs.arin.net arin-net

This will search the database for entries that contain the identifier (name,
network, host, IP number, or handle). The example searches by network name.

Special characters may be used in the identifier field to specify the search

To find only a certain TYPE of record, use keyword:

HOst
ASn
PErson
ORganization
NEtwork
GRoup

To search only a specific FIELD, use keyword or character:
HAndle or "!"
Mailbox or contains "@"
NAme or leading "."

Here are some additional Whois keywords:

EXPand or "*" Shows all parts of display without asking
Full or "=" Shows detailed display for EACH match
HElp Enters the help program for full documentation
PArtial or trailing "." Matches targets STARTING with the given string
Q, QUIT, or hit return Exits Whois

SUBdisplay or "%" Shows users of host, hosts on net, etc.
SUMmary or "$" Always shows summary, even if there is just one match

When conducting a search using the trailing "." to your input or using the PArtial
keyword, you will locate everything that starts with your input. For example,
typing "na Mack." or "na pa mack" will locate the names "Mack","MacKay",
"Mackall" etc.

To guarantee matching only a single record, look it up by its handle using a
handle-only search. For example, a search for "KH" finds all records with the
contact information for KH, but "!lKH" or "HA KH" would find only the single
record (if any) whose handle is KH . In the record summary line, the handle is
shown in parenthesis after the name, which is the first item on the line.

When using a handle to conduct a search for other information, be sure to add the
-arin extension to the handle. For example, using the handle JB2 to search the
database requires insertion of "JB2-arin" in the search field.

The Whois search program has been modified to more effectively accommodate
classless queries. Prior versions provided results on classful queries only.
To cite an example:

A query using Netnumber 10.8.0.0 under the older version of Whois yielded a "no
match found" response.

Querying 10.0.0.0, 12*, or 10. would have located up to 256 records inside the
Class A block (too much information).

Using the enhanced Whois search, the user can query any net number and locate the
network record containing the number, assuming that the number is registered
through ARIN. This is true for all classless addresses whether or not the number

```
is located at a bit boundary. Network information will be
displayed
hierarchically, with "parent," 2nd level parent, and
"children," shown in order.
```

Traceroute

To understand how traceroute works, you need a basic understanding of ICMP and ping. Let's briefly look at ping before we discuss traceroute. Ping is a program based on *Internet Control Message Protocol* (ICMP), which tells you whether a host is responding. If it is not responding, you get the following output:

```
Pinging newriders.com [10.10.10.8] with 32 bytes of data::
Request timed out.
Request timed out.
Ping statistics for 10.10.10.10:
Packets: Sent = 2, Received = 0, Lost = 2 (100% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum =  0ms, Average =  0ms Control-C
```

If a host is active on the network and responding, you get the following message:

```
Pinging 10.10.10.10 with 32 bytes of data:

Reply from 10.10.10.10: bytes=32 time=2ms TTL=255
Reply from 10.10.10.10: bytes=32 time=4ms TTL=255
Reply from 10.10.10.10: bytes=32 time=5ms TTL=255
Reply from 10.10.10.10: bytes=32 time=5ms TTL=255

Ping statistics for 10.10.10.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum =  5ms, Average =  4ms
```

Ping is useful, but in some cases, you would like to know the path a packet took through the network. In such cases, you would use a program called traceroute. Traceroute modifies the *time to live* (TTL) field to determine the path a packet takes through the network. The way TTL works is that every time a packet goes through a router, the TTL field is decremented. When a router gets a packet with a TTL of 0, it cannot forward the packet. What normally happens is when the TTL gets to 1, the current router determines whether the next hop is the destination, and if it is not, it drops the packet. Normally, it will throw the packet away and send an ICMP "time exceeded" message back to the sender. The traceroute program sends out a packet with a TTL of 1, then 2, then 3,

and so on, until it gets to the destination. This forces each router along the way to send back a time exceeded message, which can be used to track each hop from source to destination. The following is sample output from running traceroute:

```
Tracing route to [10.10.10.5]
over a maximum of 30 hops:

  1     2 ms     3 ms     3 ms  10.246.68.1
  2     4 ms     7 ms     4 ms  10.5.5.1
  3     9 ms     7 ms     7 ms  10.6.5.1
  4    12 ms     7 ms     7 ms  SOMENAME.LOCATION. NET
[10.7.1.1]
  5     8 ms    11 ms    11 ms  SOMENAME.LOCATION. NET
[10.8.1.1]
  6    11 ms    18 ms    21 ms  SOMENAME.LOCATION. NET
[10.9.1.1]
  7   120 ms    96 ms   119 ms  SOMENAME.LOCATION. NET
[10.10.1.1]
  8    82 ms   125 ms    82 ms  SOMENAME.LOCATION. NET
[10.11.1.1]
  9    97 ms    92 ms   156 ms  SOMENAME.LOCATION. NET
[10.12.1.1]
 10    81 ms    82 ms    82 ms EXTERNAL.ROUTER.LOCATION. NET
[10.13.1.1]
 11    81 ms    86 ms   108 ms   FIREWALL 10.14.1.1
 12   109 ms    85 ms    90 ms  LOCATION. NET [10.10.10.5]
Trace complete.
```

Because traceroute shows the path a packet took through a network, this information can be used to determine whether hosts are on the same network or not. Companies that are connected to the Internet have an external router that connects their networks to their ISPs or the Internet. All traffic going to a company has to go through the external router. Otherwise, there would be no way to get traffic into the network. (This is assuming that the company does not have multiple connections to the Internet.) Most companies have firewalls, so the last hop of the traceroute output would be the destination machine, the second to last hop would be the firewall, and the third to last hop would be the external router. All machines that go through the same external router are on the same network and usually belong to the same company.

By tracerouting to various IP addresses, an attacker can determine whether or not these machines are on the same network by seeing whether they went through the same external router. This can be done manually, Perl scripts could be written, or a hacker could just use the `grep` command to filter the output.

In the previous example, the 10th hop is the external router, and the 11th hop is the firewall. So now if an attacker runs several traceroutes, he can see whether or not they go through the external router, and by doing this with a bunch of addresses, he can tell which ones are on the local segment and which ones are not. So, if an attacker performs this for 10.10.10.1 and 10.10.10.5, he gets the following:

```
Tracing route to [10.10.10.5]
over a maximum of 30 hops:

  1     2 ms     3 ms     3 ms  10.246.68.1
  2     4 ms     7 ms     4 ms  10.5.5.1
  3     9 ms     7 ms     7 ms  10.6.5.1
  4    12 ms     7 ms     7 ms  SOMENAME.LOCATION. NET
[10.7.1.1]
  5     8 ms    11 ms    11 ms  SOMENAME.LOCATION. NET
[10.8.1.1]
  6    11 ms    18 ms    21 ms  SOMENAME.LOCATION. NET
[10.9.1.1]
  7   120 ms    96 ms   119 ms  SOMENAME.LOCATION. NET
[10.10.1.1]
  8    82 ms   125 ms    82 ms  SOMENAME.LOCATION. NET
[10.11.1.1]
  9    97 ms    92 ms   156 ms  SOMENAME.LOCATION. NET
[10.12.1.1]
 10    81 ms    82 ms    82 ms EXTERNAL.ROUTER.LOCATION. NET
[10.13.1.1]
 11    81 ms    86 ms   108 ms   FIREWALL 10.14.1.1
 12   109 ms    85 ms    90 ms  LOCATION. NET [10.10.10.5]

Trace complete
```

If he performs it for 10.10.9.x and 10.10.11.x, he gets the following:

```
Tracing route to [10.10.10.5]
over a maximum of 30 hops:
  1     2 ms     3 ms     3 ms  10.24.0.1
  2     4 ms     7 ms     4 ms  10.25.5.1
  3     9 ms     7 ms     7 ms  10.26.5.1
  4    12 ms     7 ms     7 ms  SOMENAME.LOCATION. NET
[10.27.1.1]
  5     8 ms    11 ms    11 ms  SOMENAME.LOCATION. NET
[10.28.1.1]
  6    11 ms    18 ms    21 ms  SOMENAME.LOCATION. NET
[10.29.1.1]
  7   120 ms    96 ms   119 ms  SOMENAME.LOCATION. NET
[10.210.1.1]
```



```
  8    82 ms   125 ms    82 ms  SOMENAME.LOCATION. NET
[10.211.1.1]
  9    97 ms    92 ms   156 ms  SOMENAME.LOCATION. NET
[10.212.1.1]
 10    81 ms    82 ms    82 ms EXTERNAL.ROUTER.LOCATION. NET
[10.213.1.1]
 11    81 ms    86 ms   108 ms   FIREWALL 10.214.1.1
 12   109 ms    85 ms    90 ms  LOCATION. NET [10.210.10.5]
Trace complete.
```

Based on the two sets of results, the attacker knows that 10.10.10.x is on the same segment or is for the same company and 10.10.x.x is not. Therefore, the range of hosts addresses are 1–254, and the subnet is 255.255.255.0.

We showed two ways that an attacker could go in and determine the range of addresses for a company. Now that an attacker has the address range, he can continue gathering information, and the next step is to find active hosts on the network.

Find Active Machines

After an attacker knows what the IP address range is, he wants to know which machines are active and which ones are not. In a lot of cases, a company gets an address range that is larger than what it needs, so it can grow into it. Also, different machines are active at different times during the day. What I have found is that if an attacker looks for active machines during the day and then again late in the evening, he can differentiate between workstations and servers. Servers should be up all the time and workstations would only be active during normal working hours.

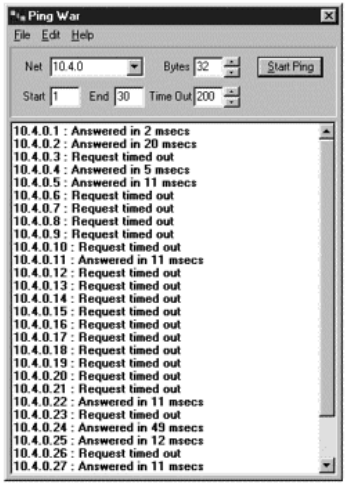
Also, because more and more companies are using *Network Address Translation* (NAT), with private addresses on the inside, this technique will sometimes provide limited information, if it is performed from the Internet. For example, if I only have two devices with external addresses and everything else is behind the firewall, an attacker might think there are only a couple of machines, when in reality there are a lot more. Thus, another benefit of using private addresses and NAT. With NAT, a company uses private addresses for its internal machines, such as the 10.x.x.x network range, and whenever these machines need to access the Internet, the device performing NAT, usually the firewall or router, translates the private address to a public address.

Ping

As we have covered, ping is a useful program for finding active machines on a network. Ping uses ICMP and works by sending an "echo request" message to a host, and if the host is not active, it does not receive a

reply, and it times out. If the host is active, then it sends back an "echo reply" to the sender of the message. Ping is a simple and straightforward way to see which machines are active and responding on a network and which ones are not. The only drawback is ping is usually used to ping one machine at a time. What an attacker would like to do is ping a large number of machines at the same time and see which ones respond. This technique is commonly referred to as *ping sweeping* because the program sweeps through a range of addresses to see which ones are active. Ping War is a useful program for finding active machines. Ping War runs on Windows machines and is available at: http://www.fantastica.com/digilex/. Ping War basically pings a range of addresses, so an attacker knows which ones are active. Figure 3.2 shows the output from Ping War:

Figure 3.2. Initial screen for Ping War.



Nmap can also be used to determine which machines are active. Nmap is a multi- purpose tool that has several features. Nmap is mainly a port scanner, but it can also be used to ping sweep an address range. Using the following syntax enables nmap to scan a range of addresses:

```
Nmap -sP -PI 10.4.0.1-30
```

The following is the output from running the command:

```
Starting nmap V. 2.53 by fyodor@insecure.org (
www.insecure.org/nmap/ )
```



```
Host 10.4.0.1 appears to be up.
Host 10.4.0.2 appears to be up.
Host 10.4.0.4 appears to be up.
Host 10.4.0.5 appears to be up.
Host 10.4.0.11 appears to be up.
Host 10.4.0.22 appears to be up.
Host 10.4.0.24 appears to be up.
Host 10.4.0.25 appears to be up.
Host 10.4.0.27 appears to be up.
```

Find Open Ports or Access Points

Now that an attacker has a pretty good map of the network and knows which machines are active and which ones are not, he can begin to assess how vulnerable the machines are. Just as a burglar would look for access points into a house to see how vulnerable it is, an attacker wants to do the same thing. In a traditional sense, the access points a thief looks for are doors and windows. These are usually the house's points of vulnerability because they are the easiest way for someone to gain access. When it comes to computer systems and networks, ports are the doors and windows of the system that an intruder uses to gain access. The more ports that are open, the more points of vulnerability, and the fewer ports, the more secure it is. Now this is just a general rule. There could be cases where a system has fewer ports open than another machine, but the ports it has open present a much higher vulnerability.

Port Scanners

To determine which ports are open on a system, an attacker would use a program called a port scanner. A port scanner runs through a series of ports to see which ones are open. There are several port scanners available, however, there are two key features that I highly recommend having in a port scanner. First, make sure it can scan a range of addresses at the same time. If you are trying to determine the vulnerabilities for your network and you have thirty machines, you are going to get really tired of scanning each machine individually. Second, make sure you can set the range of ports that the program scans for. A lot of port scanners will only scan ports 1 through 1024, or they only scan the more popular ports, which are known as well-known port numbers. This is very dangerous because, in a lot of cases, attackers know this, so if they break into your machine and open a port as a backdoor, they will open a high port, for instance 40,000, with the hope that you will not notice it. You only know every possible point of entry into a machine, if you can scan the entire range 1 through 65,535. It is also important to point out that you have to scan ports 1 through 65,525 twice—once for TCP and once for UDP. Because most companies only scan TCP, attackers like to hide on UDP ports.

There are also several different types of scans that can be performed:

- **TCP connect scan—** This is the most basic type of scan. The program tries to connect to each port on a machine using the system calls and trying to complete a three-way handshake. If the destination machine responds, then the port is active. In most cases, this type of scan works fairly well. It doesn't work if the network you are scanning is trying to hide information with a firewall or other device. Some firewalls can detect that a port scan is being hacked, and they provide limited or no information to the attacker. It also doesn't work well if you are trying to hide the fact that your are port scanning a machine. A TCP connect scan is noisy because it is easy for someone to detect, if they are watching the system.
- **TCP SYN scan—** Remember, because TCP is a reliable protocol, it uses a three-way handshake to initiate a connection. If you are trying to see whether a port is open on a machine, you would send a packet to that port with the SYN bit set. If the port is open, the machine would send back a second packet with the SYN and ACK bit set. Well, at this point, you know the port is open on the machine, and there is no need to send the third part of the three-way handshake. This technique is often referred to as having a half open connection to a machine. This type of scan is a little more stealthy than the basic scan because some machines do not log a half open connection.
- **FIN scan—** After a TCP connection is established, the two machines send packets back and forth. When they are done communicating, they send a packet with the FIN bit set, basically tearing down the connection. Well, the way TCP works is if you send a packet to a closed port, the system replies with a RST command telling you the port is not open. The way this scan works is by sending a packet with the FIN bit set. If the port is open, it ignores it, but if the port is closed, you get a RST or reset. This type of scan is very stealthy because most systems do not log these packets.
- **ACK scan—** As we have covered, to initiate a new connection, a system has to send a packet with the SYN bit set. If a system sends a packet to a machine where it does not have an active connection with the ACK bit set, and the destination machine has that port open, it will send a reset. You might be saying, "This sounds a lot like a FIN scan," but it has one big advantage. It is an easy way to get around packet filtering firewalls. Most packet filtering firewalls allow established sessions into a network. If this was not allowed, all traffic would be blocked. So, the way it is configured is if the connection is initiated from inside the network, then it allows the reply back in. The way this is done is by checking SYN and ACK flags. If the SYN bit is not set and the ACK bit is set, then the firewall assumes that it is an established session. So, doing an ACK

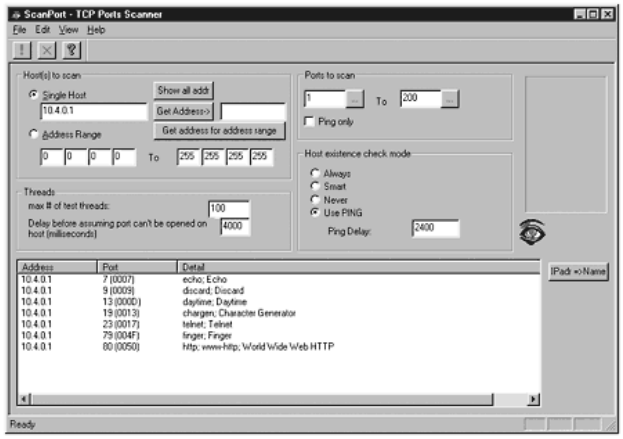
scan provides a convenient way to get around these firewalls and scan an internal host.

There are several other type of scans, but these are the most popular. Now we will take a look at port scanning programs for both the Windows and UNIX environments.

ScanPort

For a Windows environment, we are going to use a program called ScanPort. It is a fairly basic port scanner, but it enables you to specify both a range of addresses and range of ports to scan. ScanPort is written by DataSet and is available at: http://www.dataset.fr/eng/scanport.html. Figure 3.3 is the output from running ScanPort against a single machine.

Figure 3.3. Running ScanPort on a Windows machine.



In this case, it was a web server that the administrator told me only had port 80 open. It is pretty interesting what you will find when you start port scanning machines.

Nmap

On the UNIX side, the port scanner that I recommend is nmap. Nmap is much more than a port scanner, and it is a necessary tool for your security toolbox. Nmap enables you to run all the different types of scans we talked about and has a lot of other useful features. The following is the output from nmap:

```
Starting nmap V. 2.53 by fyodor@insecure.org (
www.insecure.org/nmap/ )
Interesting ports on  (10.246.68.1):
(The 1516 ports scanned but not shown below are in state:
closed)
Port       State       Service
7/tcp      open        echo
9/tcp      open        discard
13/tcp     open        daytime
19/tcp     open        chargen
23/tcp     open        telnet
79/tcp     open        finger
80/tcp     open        http
Nmap run completed -- 1 IP address (1 host up) scanned in 2
seconds
```

After running the port scanners, an attacker has a really good idea of the access points into the computer systems.

War Dialing

Another common access point into a network is modems. You do not know how many times I have been performing a security assessment where the company had very good Internet security. They had a properly configured firewall and minimal access, but they broke the cardinal rule that all traffic in and out of your network must go through the firewall. They had the modem pool and random modems connected to servers that were behind the firewall. This meant once I was able to locate the modems, I could dial-in to try to crack the passwords, and in several cases, there were no passwords.

Programs for finding modems on a network are called *war dialers*. Basically, you put in the starting numbers or the range of phone numbers you want it to scan, and it will dial each number looking for a modem to answer, and if a modem answers, then it records this information.

THC-Scan

Several war dialers are available on both shareware and commercial, but the one we will cover is THC-Scan. THC-Scan runs in a DOS window in a Windows environment. Figure 3.4 is the main screen for THC-Scan.

Figure 3.4. THC-Scan's main screen.

THC–Scan has most of the features an attacker would need to perform war dialing tasks. Some of these features are:

- Support for both carrier and tone mode
- Variable dialing features. This enables the program to dial the numbers in sequential or random order.
- Distributed feature that enables various machines or modems to work together.
- Jamming detection, if it starts to detect a high number of busy signals
- Random wait between calls

As you can see, the program has several features to accomplish war dialing. The key thing to emphasize with war dialing is that the program actually rings every phone and waits for someone to answer. If a person answers, it disconnects, but if a modem answers, it records the information and then disconnects. An attacker could also set the program to connect if a modem answers, at which point, it tries to determine what program is running, and in some cases, it even tries to guess the password. This is important to point out because if an attacker performs war dialing in sequential order, a company would see one phone after another ring, and when the person picks up, no one is there. This would look very suspicious, and this is why war dialing is usually done after hours—to minimize the chance of detection.

Figure Out the Operating System

Now that the attacker is starting to make a lot of progress—he knows which machines are active and which ports are open—it would be useful for him to identify which operating system each host is running. There are programs that probe the remote hosts to determine which operating system is running. This is done by sending the remote host unusual packets or packets that do not make sense. Because these packets are

not specified in the RFC, each operating system handles them differently, and by parsing the output, the attacker can figure out what type of device he is accessing and which operating system (OS) is running. Just to give an example, one type of packet used is a packet with the SYN and FIN bits both set. In normal operations, this type of packet should not occur, so when the operating system responds to this packet, it does so in a predictable fashion, which enables the program to determine which operating system the host is running. Also, the sequence numbers used with TCP have various levels of randomness, depending on which operating system is running. The programs also use this information to make a best guess at what the remote OS is.

Queso

Queso is the original program that performs this function. Queso currently identifies around 100 different devices ranging from Microsoft to UNIX to Cisco routers. As you can see, this is a great tool that will help an attacker figure out the target OS, so he can focus in on the OS to compromise it. The following is the output from running queso against an IP address:

```
10.246.68.1:80     * Cisco 11.2(10a), HP/3000 DTC, BayStack
Switch
```

As you can see, it correctly identified the device as a Cisco router. Now, from a security standpoint, you would make sure that all the proper patches have been applied, so the device cannot be compromised. I have also known cases where administrators have changed some of the default behavior on these devices to try to fool these programs.

Nmap

The other program that enables you to do this is nmap. It has the same functionality as queso, I just prefer it because it is an all-in-one tool and has additional features. It can also detect more devices. Currently, it can detect close to 400 different devices. The following is the output from running nmap with the OS fingerprinting option turned on:

```
Starting nmap V. 2.53 by fyodor@insecure.org (
www.insecure.org/nmap/ )
Interesting ports on  (10.246.68.1):
(The 1516 ports scanned but not shown below are in state:
closed)
Port       State       Service
7/tcp      open        echo
9/tcp      open        discard
13/tcp     open        daytime
19/tcp     open        chargen
```



```
23/tcp     open        telnet
79/tcp     open        finger
80/tcp     open        http

TCP Sequence Prediction: Class=random positive increments
                         Difficulty=2489 (Medium)
Remote operating system guess: Cisco IOS 11.3 - 12.0(9)

Nmap run completed -- 1 IP address (1 host up) scanned in 2
seconds

Starting nmap V. 2.53 by fyodor@insecure.org (
www.insecure.org/nmap/ )
Interesting ports on  (208.246.68.48):
(The 1508 ports scanned but not shown below are in state:
closed)
Port       State       Service
21/tcp     open        ftp
23/tcp     open        telnet
25/tcp     open        smtp
79/tcp     open        finger
98/tcp     open        linuxconf
111/tcp    open        sunrpc
113/tcp    open        auth
513/tcp    open        login
514/tcp    open        shell
515/tcp    open        printer
948/tcp    open        unknown
1024/tcp   open        kdm
1025/tcp   open        listen
1032/tcp   open        iad3
6000/tcp   open        X11

TCP Sequence Prediction: Class=random positive increments
                         Difficulty=920729 (Good luck!)
Remote operating system guess: Linux 2.1.122 - 2.2.14

Nmap run completed -- 1 IP address (1 host up) scanned in 1
second

Starting nmap V. 2.53 by fyodor@insecure.org (
www.insecure.org/nmap/ )
Interesting ports on  (208.246.68.40):
(The 1522 ports scanned but not shown below are in state:
closed)
Port       State       Service
139/tcp    open        netbios-ssn

TCP Sequence Prediction: Class=trivial time dependency
                         Difficulty=1 (Trivial joke)
```

```
Remote operating system guess: Windows NT4 / Win95 / Win98

Nmap run completed -- 1 IP address (1 host up) scanned in 3
seconds
```

In this example, an attacker ran nmap against three devices, one was a Cisco router, one was a Linux machine, and one was a Windows 98 machine. All were correctly identified.

Figure Out Which Services Are Running on Each Port

Now that an attacker knows which operating system is running, the IP address, and which ports are open, the attacker needs to find out which services are running on each port. Knowing which specific service is running enables the attacker to look up exploits and launch known vulnerabilities against the service. The first way to do this is to utilize the default information.

Default Port and OS

Based on common configuration and software, the attacker can make a best guess of what services are running on each port. For example, if he knows that the operating system is a UNIX machine and port 25 is open, he can assume it is running sendmail, and if the operating system is Microsoft NT and port 25 is open, he can assume it is running Exchange. This is an easy way to figure out which service is running, however we do not have the details an attacker wants, for example, which version of the software. Also, just because port 25 is open does not mean it is running a mail program. On most systems it is, but it is not guaranteed. A more accurate way to obtain this information is with a manual method.

Telnet

Telnet is a program that comes with most operating systems that enables you to connect to a specific port on a destination machine. We will cover other programs, such as netcat, which also enable you to do this. With these programs, an attacker would connect to the port that is open and would hit the enter key a couple of times. The default installation of most operating systems displays banner information about what services are running on a given port. The following is an example of connecting to two different ports on a Linux system:

- Connecting to port 25:

```
 Red Hat Linux release 6.2 (Zoot)
 Kernel 2.2.14-5.0smp on an i686
login:
```


- Port 25 (telnet 10.10.10.5 25):

```
 220 linux1 ESMTP Sendmail 8.9.3/8.9.3;
Wed, 27 Dec 2000 21:32:55 -0500
```

As you can see, the system tells you not only what service is running, but what version and what the underlying operating system is. A company giving this information away is just making it way to easy for an attacker. As much as possible, this information needs to be removed or sanitized before an operation system goes live.

Vulnerability Scanners

Vulnerability scanners are programs that can be run against a site that give a hacker a list of vulnerabilities on the target host. The following are several different vulnerability scanners that are currently available:

- Commercial:
  - ISS's Internet Scanner (http://www.iss.net)
  - Network Associates' CyberCop Scanner (http://www.pgp.com/products/cybercop-scanner/default.asp)
  - Cisco's Secure Scanner (formerly NetSonar) (http://www.cisco.com/warp/public/cc/pd/sqsw/nesn/)
  - Axent's NetRecon (http://www.axent.com)

- Shareware:
  - SARA, by Advanced Research Organization (http://www-arc.com/sara/)
  - SAINT, by World-wide Digital Security (http://www.wwdsi.com/saint/)
  - VLAD the Scanner, by Razor (http://razor.bindview.com/tools/)
  - Nessus, by the Nessus Project Team (http://www.nessus.org)

This is not a comprehensive list, however it is meant to give you an idea of the programs available. Because this chapter is on information gathering, these programs will not be covered in depth. They are mentioned because many of the vulnerability scanners will try to probe each port to verify or figure out which service is running. In my experience, they are not always as detailed or as accurate as the manual method of telneting to each port, but they are a lot quicker.

Map Out the Network

Now that an attacker has gained all this information, he wants to map out your network, so he can figure out the best way to break in. When a thief is going to rob a bank, what does he do? He either acquires the blueprints for the building or he visits the building and draws a map of the floor plan.

This way, he can figure out the best way to successfully pull off his robbery. To do this with a network, there are manual and automatic ways to determine this information. We will briefly show how an attacker can use traceroute or ping to find out the information. He could also use a program such as cheops, which automatically maps the network for him.

Traceroute

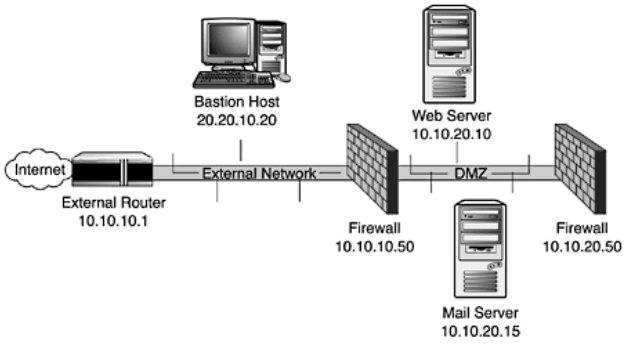
As we already discussed, traceroute is a program that can be used to determine the path from source to destination. By combining this information, an attacker determines the layout of a network and the location of each component.

For example, after running several traceroutes, an attacker might obtain the following information:

- traceroute 10.10.10.20, second to last hop is 10.10.10.1
- traceroute 10.10.20.10, third to last hop is 10.10.10.1
- traceroute 10.10.20.10, second to last hop is 10.10.10.50
- traceroute 10.10.20.15, third to last hop is 10.10.10.1
- traceroute 10.10.20.15, second to last hop is 10.10.10.50

By putting this information together, he can diagram the network, as shown in Figure 3.5.

Figure 3.5. Diagram of sample network an attacker was able to map out using traceroute.

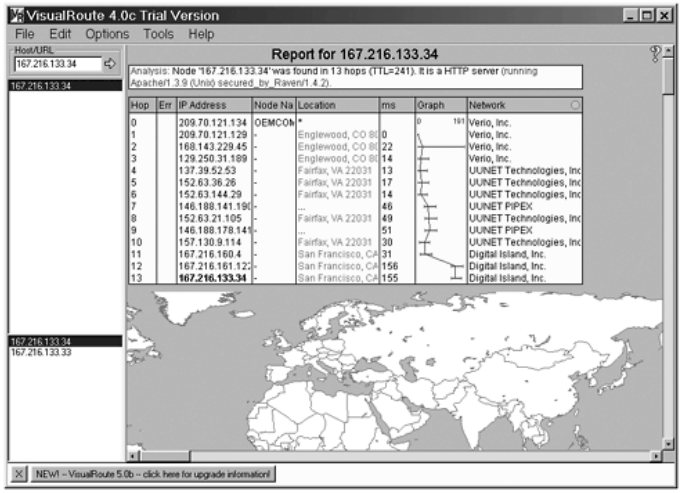


Visual Ping

To show you the power of such techniques, let's start to utilize some programs that help automate this process. VisualRoute is a program that visually shows the route a packet took through the Internet. Not only does it show an attacker the systems it went through, but it also shows an

attacker where the system is located geographically. Figure 3.6 shows an example of running VisualRoute.

Figure 3.6. Example of using VisualRoute to identify the location of a machine.

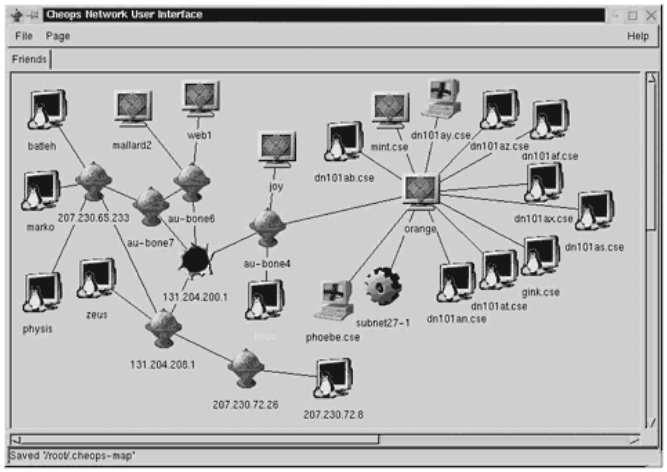


By running this multiple times against several hosts, an attacker can get a good idea of whether two systems are on the same network. This is only a little more automated than the manual method, so now let's look at a program that automates the entire process.

Cheops

Cheops utilizes the techniques just mentioned to map out a network and display a graphical representation of the network. Now, if this is run from the Internet, it is only able to map out the portion of the network that it has access to. So, any machine that is not accessible from the Internet, such as non-routable addresses, are not able to be mapped. Thus, another reason to use non-routable addresses whenever possible. Figure 3.7 is sample output from running cheops.

Figure 3.7. Output from running cheops against a sample network.



Cheops is basically a network mapping tool, but if a company is not careful, it can be used against it. Cheops not only maps a network, but it performs operating system fingerprinting to determine what the operating system is on a given system, and it displays it with the appropriate icon. As you can see, these programs are getting more and more powerful—a single program can perform multiple functions, which makes it much easier for an attacker. This means that companies must take the time to properly secure their networks.

Information Gathering Summary

So far in this Chapter, we have covered the steps that an attacker would take to gather information about a company or the steps you would take to see what information is available about your network, so that you can secure your system. It is important to remember that someone cannot just directly attack your system. They have to spend some time gathering information, so they know what they are attacking. If a company could limit what information it gives out, it would not only make it harder for someone to attack its system, but it would make it less of a target. In this current environment, there are so many systems with no or minimal security that if your site looks harder to get into, there is a good chance an attacker will pass you by. The important thing to remember is that the earlier you can detect someone doing damage, the better off you are. So, by understanding the steps an attacker would take to gather information on your site, the better chance you have of detecting him and stopping him before he causes more damage.

To better illustrate the information gathering process, we will cover an example of red teaming. In a lot of cases, especially with companies that

have sensitive operations, such as banks, the company wants to know the threat it has to external attackers and the points of vulnerability without giving away any information. Basically, these companies want to simulate an attack with a trusted entity that will then help them improve their security. This process is often referred to as red teaming because it provides insight into the steps an attacker would take to compromise your system. The following section illustrates the previous steps we have covered in an example where we compromise a fictitious company. So, let's put together our red team and start gaining information about our target network.

Red Teaming

In the first half of this Chapter, we went over the steps that an attacker would perform, but we went over each step independently. What makes these steps so powerful is when you combine them together to see the end result. To help illustrate this, we will go through an example of how a hacker would perform this type of attack against your company. Because the real interest is for you to understand it, so you can perform it against your company with the goal of securing your system, we will call in a red team exercise. We will also look at what can be done to minimize the amount of information someone can gather from your site. To do this, each step will be followed by a section called "Protection", which will tell you what can be done to minimize the impact to your company. I recommend performing these steps against your company, and after you determine your points of vulnerability, follow the procedures on how to protect against them. Remember, always under any circumstance, get written permission before installing or running these tools against a network!

In this example, we are going to go after a fictitious company, company X. The following are the basic steps we are going to cover:

- Whois
- NsLookup
- ARIN Web Search
- Traceroute
- Ping
- Map the network
- PortScan and Fingerprinting
- Exploiting the System

Whois

Now that we have decided to target Company X, the first thing we want to do is perform a whois lookup on its domain name to find out additional information. The following is the output from whois:

```
*** Connecting to whois.networksolutions.com
*** Connected established

The Data in Network Solutions' WHOIS database is provided by
Network
Solutions for information purposes, and to assist persons in
obtaining information
about or related to a domain name registration record.
Network Solutions does not guarantee its accuracy.  By
submitting a WHOIS query,
you agree that you will use this Data only for lawful purposes
and that, under no
circumstances will you use this Data to: (1) allow, enable, or
otherwise support
the transmission of mass unsolicited, commercial advertising
or solicitations via
e-mail (spam); or  (2) enable high volume, automated,
electronic processes that
apply to Network Solutions (or its systems).  Network
Solutions reserves the right
to modify these terms at any time.  By submitting this query,
you agree to abide
by this policy.

Registrant:
Eric Test (TESTDOMAIN-DOM)
   21225 Somewhere Drive
   Somewhere, SW 22534
   US

   Domain Name: TESTCOMPANYX DOMAIN

   Administrative Contact, Technical Contact, Zone Contact,
Billing Contact:
      Cole, Eric  (EC2515)  ERIC@AYCE.COM
Eric Test (TESTDOMAIN-DOM)
   21225 Somewhere Drive
   Somewhere, SW 22534
   US
   555-555-5555 fax 444-444-4444
   Record last updated on 22-Jul-1999.
   Record expires on 17-Apr-2001.
   Record created on 17-Apr-1998.
   Database last updated on 27-Jul-2000 06:19:54 EDT.
   Domain servers in listed order:

   MAIL2.TESTDOMAIN     10.196.0.38
   MAIL1.TESTDOMAIN     10.45.32.38
```



```
*** Connection closed
```

As we stated earlier in this chapter, we get a lot of important information, but the pieces we care about are the two domain name servers. If we were performing a social engineering attack, then we would use the other information.

Protection

There are a couple of things you can do to minimize the potential damage the whois lookup can cause. Remember that you need to have a domain record and that data has to be somewhat valid because that is what people use to contact you or your company. The first piece of information is the contact information. I recommend putting a position title with a general number, as opposed to with a specific person, so the potential for social engineering is reduced. The other thing you can do is list your number, but make up a fictitious name and email. You would check the email and phone calls for this person, but from a social engineering standpoint, if anyone calls up asking for this person or throwing their name around, it should set off an immediate flag. This is a good way to turn the tables on the attacker and trap him. Be very careful because I have seen cases where companies try to out smart an attacker, and it backfires on them.

To protect against the DNS problem, run your own DNS server with split DNS. This way, an attacker queries the external DNS server where he only gets a minimal amount of information.

Nslookup

Now that we have the names of the DNS servers, we want to use nslookup to try to find the IP address of some servers. Remember, the server we are connecting to is authoritative for the domain we are after, so it will have records listed for the mail server, the web server, and possibly other servers. The following is the output for our nslookup:

```
Default Server:  companyx test domain
Address:  10.246.68.129

> Server:  companyx test domain
Address:  10.246.68.129

> [companyx test domain]
 company x                      NS      server =
firewall.air.org
 qaprogram                    A       10.246.68.155
 localhost                      A       127.0.0.1
```

```
 lists                                  A      10.246.68.132
 gate2                                A      10.246.68.140
 ip                                     A      10.246.68.157
 mail                                 A      10.246.68.50
 idea                                 A      10.246.68.139
 randd                              A      10.246.68.37
 project y                          A      10.246.68.138
 motor                              A      10.246.68.141
 et                                     A      10.246.68.35
 firewall                           A      10.246.68.129
 secure                             A      10.246.68.156
 cef                                   A      10.237.183.73
 cep                                  A      10.246.68.131
 oda1                                A      10.246.68.136
 oda2                                A      10.246.68.42
 ip2                                   A      10.246.68.137
 www                              A      10.246.68.133
 seagate-info                   A      10.246.68.55
 mail                                A      10.246.200.91
 lists2                              A      10.246.68.144
>
```

I issued the command `server=DNS server` to set the system to the authoritative DNS server, and then I issued an `ls` command followed by the domain name to get a list of the servers. (Remember, to protect the innocent, I changed all the valid IPs to the 10.x.x.x network.) Now we have a range of IP addresses we can use to try to find out the address space this company has.

Protection

Certain records have to appear in your DNS records, but you should minimize the amount that occurs. The less information you give out the better. Second, any IP address listed should be statically mapped through a firewall with only a specific port allowed through. For example, your mail server should be behind a firewall with a non-routable address. The firewall would then have a static mapping, which means anyone who is trying to get to this address is automatically mapped to the mail server's private address and is only allowed through on port 25, nothing else. This will minimize your exposure and will help protect the system.

ARIN Web Search

Now we want to try and figure out the network address and subnet. Remember, there are two ways to do this, but the easiest is using ARIN. So, in our web browser, we would go to http://www.arin.net/whois and put in one of the IPs to see if we get a hit. So, we put in **`10.246.69.139`** , and we get the following output:


```
SOME ISP PROVIDER, Inc. (NETBDNS-1996B) JDJKS996B
     10.249.255.255
ISP/COMPANY X (NETB-DH-10-246-68)     10-246-68
     10.246.68.0 -
     20.146.68.255
```

This tells us a lot of information. We know that the address class 10.249 belongs to the ISP, but the company we are interested in only has 10.246.68, which means it has 254 possible machines on the network, unless it is performing NAT.

Protection

With ARIN, there is not a lot you can do except to make sure you only use these addresses for external devices, such as routers and firewalls. Any other device should use a private address and should be behind a firewall. This will limit the value of the information and the potential damage an attacker can cause.

Traceroute

Because we obtained the information we needed from an ARIN search, traceroute is not necessary, but let's perform some tests anyway just to confirm our results. First, let's perform a traceroute to 10.246.68.144 because we know it is a valid address. When we do this, we get the following results:

```
Tracing route to [10.10.10.5]
over a maximum of 30 hops:

  1     2 ms     3 ms     3 ms  10.246.68.1
  2     4 ms     7 ms     4 ms  10.5.5.1
  3     9 ms     7 ms     7 ms  10.6.5.1
  4    12 ms     7 ms     7 ms  SOMENAME.LOCATION. NET
[10.7.1.1]
  5       ms    11 ms    11 ms  SOMENAME.LOCATION. NET
[10.8.1.1]
  6    11 ms    18 ms    21 ms  SOMENAME.LOCATION. NET
[10.9.1.1]
  7   120 ms    96 ms   119 ms  SOMENAME.LOCATION. NET
[10.10.1.1]
  8    82 ms   125 ms    82 ms  SOMENAME.LOCATION. NET
[10.11.1.1]
  9    97 ms    92 ms   156 ms  SOMENAME.LOCATION. NET
[10.12.1.1]
 10    81 ms    82 ms    82 ms EXTERNAL.ROUTER.LOCATION. NET
[10.13.1.1]
```

```
 11    81 ms    86 ms   108 ms   FIREWALL 10.14.1.1
 12   109 ms    85 ms    90 ms  LOCATION. NET [10.248.68.144]

Trace complete.
```

Now we know the address for the external router and firewall. All traffic going to this network has to go through this router, unless it has a second connection. If it did have a second connection, we would see the other external router address when we ran traceroutes to other addresses and it would record that also. In this case, let's assume a single connection to the Internet. Now, let's run a trace to 10.246.68.1 to see the range of addresses it has:

```
Tracing route to [10.10.10.5]
over a maximum of 30 hops:

  1     2 ms     3 ms     3 ms  10.246.68.1
  2     4 ms     7 ms     4 ms  10.5.5.1
  3     9 ms     7 ms     7 ms  10.6.5.1
  4    12 ms     7 ms     7 ms  SOMENAME.LOCATION. NET
[10.7.1.1]
  5     8 ms    11 ms    11 ms  SOMENAME.LOCATION. NET
[10.8.1.1]
  6    11 ms     8 ms    21 ms  SOMENAME.LOCATION. NET
[10.9.1.1]
  7   120 ms    96 ms   119 ms  SOMENAME.LOCATION. NET
[10.10.1.1]
  8    82 ms   125 ms    82 ms  SOMENAME.LOCATION. NET
[10.11.1.1]
  9    97 ms    92 ms   156 ms  SOMENAME.LOCATION. NET
[10.12.1.1]
 10    81 ms    82 ms    82 ms EXTERNAL.ROUTER.LOCATION. NET
[10.13.1.1]
 11    81 ms    86 ms   108 ms   FIREWALL 10.14.1.1
 12   109 ms    85 ms    90 ms  LOCATION. NET [10.246.68.1]

Trace complete.
```

Let's also trace to 10.246.68.254:

```
Tracing route to [10.10.10.5]
over a maximum of 30 hops:

  1     2 ms     3 ms     3 ms  10.246.68.1
  2     4 ms     7 ms     4 ms  10.5.5.1
  3     9 ms     7 ms     7 ms  10.6.5.1
  4    12 ms     7 ms     7 ms  SOMENAME.LOCATION. NET
[10.7.1.1]
```



```
  5     8 ms    11 ms    11 ms  SOMENAME.LOCATION. NET
[10.8.1.1]
  6    11 ms    18 ms    21 ms  SOMENAME.LOCATION. NET
[10.9.1.1]
  7   120 ms    96 ms   119 ms  SOMENAME.LOCATION. NET
[10.10.1.1]
  8    82 ms   125 ms    82 ms  SOMENAME.LOCATION. NET
[10.11.1.1]
  9    97 ms    92 ms   156 ms  SOMENAME.LOCATION. NET
[10.12.1.1]
 10    81 ms    82 ms    82 ms EXTERNAL.ROUTER.LOCATION. NET
[10.13.1.1]
 11    81 ms    86 ms   108 ms   FIREWALL 10.14.1.1
 12   109 ms    85 ms    90 ms  LOCATION. NET [10.246.68.254]

Trace complete.
```

By analyzing the results, we now see that they have the entire last octet. Now we need to see if they also have all or some of the second octet. If we trace to anything in 10.246.x, we get the following results:

```
Tracing route to [10.10.10.5]
over a maximum of 30 hops:

  1     2 ms     3 ms     3 ms  10.246.68.1
  2     4 ms     7 ms     4 ms  10.5.5.1
  3     9 ms     7 ms     7 ms  10.6.5.1
  4    12 ms     7 ms     7 ms  SOMENAME.LOCATION. NET
[20.7.1.1]
  5     8 ms    11 ms    11 ms  SOMENAME.LOCATION. NET
[20.8.1.1]
  6    11 ms    18 ms    21 ms  SOMENME.LOCATION. NET
[20.9.1.1]
  7   120 ms    96 ms   119 ms  SOMENAME.LOCATION. NET
[20.10.1.1]
  8    82 ms   125 ms    82 ms  SOMENAME.LOCATION. NET
[20.11.1.1]
  9    97 ms    92 ms   156 ms  SOMENAME.LOCATION. NET
[20.12.1.1]
 10    81 ms    82 ms    82 ms EXTERNAL.ROUTER.LOCATION. NET
[20.13.1.1]
 11    81 ms    86 ms   108 ms   FIREWALL 20.14.1.1
 12   109 ms    85 ms    90 ms  LOCATION. NET [10.246.x.x]

Trace complete.
```

Because these traces go to a totally different location, this shows us that none of these addresses belong to the company and that its address space

is 20.246.68.x. Now we know the range of its network and can finish mapping it out.

Protection

Traceroute is hard to protect against because if you disable ICMP traffic, which is what traceroute uses, you loose a valuable troubleshooting tool. Once again, using private addresses inside your firewall limits the machines to which an attacker could traceroute. You could block ICMP traffic at your external router, which would help with this problem, but this would severely limit your ability as an administrator. Remember, even if we did not use traceroute, we still received the information we needed from ARIN.

Remember to enforce a principle of least privilege on your systems and network. Give entities the access they need to do their job and nothing else. If it is critical for people to have the ability to run external traceroutes, then you might not be able to disable it. On the other hand, if it is not needed, then it should be disabled.

Ping

At this point, we know what addresses belong to Company X, and we want to see what machines are active. The easiest way to do this is to ping the entire range of addresses and see which ones respond. When we run the ping at 2:00 in the morning, we get the following results (to conserve space, we will only show the results for the first 50 machines):

```
10.246.68.1 : Answered in 3 msecs
10.246.68.2 : Answered in 21 msecs
10.246.68.3 : Answered in 7 msecs
10.246.68.4 : Answered in 7 msecs
10.246.68.5 : Answered in 11 msecs
10.246.68.6 : Answered in 37 msecs
10.246.68.7 : Answered in 73 msecs
10.246.68.8 : Answered in 27 msecs
10.246.68.9 : Answered in 17 msecs
10.246.68.10 : Answered in 71 msecs
10.246.68.11 : Request timed out
10.246.68.12 : Request timed out
10.246.68.13 : Request timed out
10.246.68.14 : Request timed out
10.246.68.15 : Request timed out
10.246.68.16 : Request timed out
10.246.68.17 : Request timed out
10.246.68.18 : Request timed out
10.246.68.19 : Request timed out
10.246.68.20 : Request timed out
```



```
10.246.68.21 : Request timed out
10.246.68.22 : Request timed out
10.246.68.23 : Request timed out
10.246.68.24 : Request timed out
10.246.68.25 : Request timed out
10.246.68.26 : Request timed out
10.246.68.27 : Request timed out
10.246.68.28 : Request timed out
10.246.68.29 : Request timed out
10.246.68.30 : Request timed out
10.246.68.31 : Request timed out
10.246.68.32 : Request timed out
10.246.68.33 : Request timed out
10.246.68.34 : Request timed out
10.246.68.35 : Request timed out
10.246.68.36 : Request timed out
10.246.68.37 : Request timed out
10.246.68.38 : Request timed out
10.246.68.39 : Request timed out
10.246.68.40 : Request timed out
10.246.68.41 : Request timed out
10.246.68.42 : Request timed out
10.246.68.43 : Request timed out
10.246.68.44 : Request timed out
10.246.68.45 : Request timed out
10.246.68.46 : Request timed out
10.246.68.47 : Request timed out
10.246.68.48 : Request timed out
10.246.68.49 : Request timed out
10.246.68.50 : Request timed out
```

We then ran it at 2:00 in the afternoon and received the following results:

```
10.246.68.1 : Answered in 3 msecs
10.246.68.2 : Answered in 21 msecs
10.246.68.3 : Answered in 7 msecs
10.246.68.4 : Answered in 7 msecs
10.246.68.5 : Answered in 11 msecs
10.246.68.6 : Answered in 37 msecs
10.246.68.7 : Answered in 73 msecs
10.246.68.8 : Answered in 27 msecs
10.246.68.9 : Answered in 17 msecs
10.246.68.10 : Answered in 71 msecs
10.246.68.11 : Answered in 10 msecs
10.246.68.12 : Request timed out
10.246.68.13 : Request timed out
10.246.68.14 : Answered in 17 msecs
10.246.68.15 : Answered in 17 msecs
10.246.68.16 : Request timed out
```

```
10.246.68.17 : Request timed out
10.246.68.18 : Answered in 17 msecs
10.246.68.19 : Request timed out
10.246.68.20 : Request timed out
10.246.68.21 : Answered in 12 msecs
10.246.68.22 : Answered in 12 msecs
10.246.68.23 : Request timed out
10.246.68.24 : Request timed out
10.246.68.25 : Answered in 11 msecs
10.246.68.26 : Answered in 32 msecs
10.246.68.27 : Answered in 11 msecs
10.246.68.28 : Request timed out
10.246.68.29 : Request timed out
10.246.68.30 : Answered in 10 msecs
10.246.68.31 : Request timed out
10.246.68.32 : Answered in 12 msecs
10.246.68.33 : Answered in 20 msecs
10.246.68.34 : Request timed out
10.246.68.35 : Request timed out
10.246.68.36 : Request timed out
10.246.68.37 : Answered in 14 msecs
10.246.68.38 : Answered in 8 msecs
10.246.68.39 : Answered in 11 msecs
10.246.68.40 : Answered in 8 msecs
10.246.68.41 : Request timed out
10.246.68.42 : Answered in 15 msecs
10.246.68.43 : Answered in 12 msecs
10.246.68.44 : Request timed out
10.246.68.45 : Answered in 16 msecs
10.246.68.46 : Answered in 11 msecs
10.246.68.47 : Answered in 15 msecs
10.246.68.48 : Answered in 11 msecs
10.246.68.49 : Answered in 8 msecs
10.246.68.50 : Answered in 15 msecs
```

What this tells us is that we have a really good idea that the IP addresses 10.246.68.1 through 10.246.68.10 are servers, and the remaining addresses are client machines. This was determined by the fact that only servers should be active late at night and workstations should be active during the day. This is important information because depending on what an attacker is trying to do, he might want to go after a certain type of machine. If he wanted to install a backdoor on a machine, so he could access it late at night, but it is a user's machine that gets shut off, then it does not help him very much. So, in this case, an attacker might want to target a server instead.

Protection

Ping is hard to protect against because if you disable ICMP traffic, which is what ping uses, you loose a valuable troubleshooting tool. Once again,

using private addresses inside your firewall limits the machines an attacker could ping. You could block ICMP traffic at your external router, which would limit the information an attacker could obtain, but this would severely limit your ability as an administrator.

Map the Network

At this point, we can map out the network because we know which machines are located where and which machines are active. After the next couple of steps, we can fill in the missing pieces—what ports are open and what operating systems are being run. We could also use a mapping program, such as cheops, to validate the information we have already obtained.

PortScan and Fingerprinting

At this point, we know which machines are active and which ones are servers. Now we would like to know what operating systems are being run and what ports are open. With that information, we can target a host with a specific exploit. We can kill two birds with one stone by running nmap with the –O option. This will give us the operating system and the open ports. Here is the output from running it on the first 5 IP addresses:

```
Starting nmap V. 2.53 by fyodor@insecure.org (
www.insecure.org/nmap/ )
Interesting ports on  (10.4.0.1):
(The 1516 ports scanned but not shown below are in state:
closed)
Port       State       Service
7/tcp      open        echo
9/tcp      open        discard
13/tcp     open        daytime
19/tcp     open        chargen
23/tcp     open        telnet
79/tcp     open        finger
80/tcp     open        http

TCP Sequence Prediction: Class=random positive increments
                         Difficulty=368 (Medium)
Remote OS guesses: Cisco IOS 11.3 - 12.0(9), Cisco IOS
v11.14(CA)/12.0.2aT1/v12.0.3T

Nmap run completed -- 1 IP address (1 host up) scanned in 3
seconds

Starting nmap V. 2.53 by fyodor@insecure.org (
www.insecure.org/nmap/ )
Interesting ports on  (10.4.0.2):
```

```
(The 1520 ports scanned but not shown below are in state:
closed)
Port       State       Service
7/tcp       pen        echo
9/tcp      open        discard
19/tcp     open        chargen

TCP Sequence Prediction: Class=random positive increments
                         Difficulty=2465249 (Good luck!)
Remote operating system guess: NetWare 4.11 SP8a - Netware 5
SP4

Nmap run completed -- 1 IP address (1 host up) scanned in 6
seconds

Starting nmap V. 2.53 by fyodor@insecure.org (
www.insecure.org/nmap/ )
Interesting ports on  (10.4.0.3):
(The 1520 ports scanned but not shown below are in state:
closed)
Port       State       Service
23/tcp     open        telnet
79/tcp     open        finger
80/tcp     open        http

TCP Sequence Prediction: Class=random positive increments
                         Difficulty=1833 (Medium)
Remote OS guesses: Cisco IOS 11.3 - 12.0(9), Cisco IOS
v11.14(CA)/12.0.2aT1/v12.0.3T

Nmap run completed -- 1 IP address (1 host up) scanned in 3
seconds

Starting nmap V. 2.53 by fyodor@insecure.org (
www.insecure.org/nmap/ )
Interesting ports on  (10.4.0.4):
(The 1507 ports scanned but not shown below are in state:
closed)
Port       State       Service
25/tcp     open        smtp
27/tcp     open        nsw-fe
42/tcp     open        nameserver
80/tcp     open        http
110/tcp    open        pop-3
119/tcp    open        nntp
135/tcp    open        loc-srv
139/tcp    open        netbios-ssn
143/tcp    open        imap2
389/tcp    open        ldap
443/tcp    open        https
```



```
563/tcp    open        snews
593/tcp    open        http-rpc-epmap
636/tcp    open        ldapssl
993/tcp    open        imaps
995/tcp    open        pop3s

TCP Sequence Prediction: Class=trivial time dependency
                         Difficulty=2 (Trivial joke)
Remote operating system guess: Windows NT4 / Win95 / Win98
Nmap run completed -- 1 IP address (1 host up) scanned in 2
seconds

Starting nmap V. 2.53 by fyodor@insecure.org (
www.insecure.org/nmap/ )
Interesting ports on  (10.4.0.5):
(The 1514 ports scanned but not shown below are in state:
closed)
Port       State       Service
21/tcp     open        ftp
80/tcp     open        http
135/tcp    open        loc-srv
139/tcp    open        netbios-ssn
443/tcp    open        https
1032/tcp   open        iad3
1521/tcp   open        ncube-lm
1526/tcp   open        pdap-np
1723/tcp   open        pptp

TCP Sequence Prediction: Class=trivial time dependency
                         Difficulty=2 (Trivial joke)
Remote operating system guess: Windows NT4 / Win95 / Win98

Nmap run completed -- 1 IP address (1 host up) scanned in 3
seconds
```

Now we can see that we have two Cisco devices, one Netware and two Windows machines.

Protection

Once again, the best means of protection is a firewall that properly blocks traffic and only allows traffic on specific ports to specific machines. This way, the attacker only gets a limited view of what is going on. Remember the less information an attacker has the better.

Exploiting the System

At this point, we have a really clear map of the network, active machines, type of machines, and potential vulnerabilities. Now it is just a matter of exploiting those machines. The way this is usually done is after you know

the operating system, version, and open ports, you look up known vulnerabilities in a database or on the Internet and go after those first. Exploiting systems is what this book is about. So as we go through this book, covering each exploit and how they work, remember this section and how it fits into the big picture

Summary

This chapter laid the groundwork for the steps an attacker would take to plan an attack. It also gave a roadmap for the rest of this book. Everything else that we cover fits into this picture. It is always important to remember that the sooner you can stop someone by limiting the information they gain or the sooner you can detect someone trying to get into your system, the more secure your network will be. The other key point is that even though what we covered in this chapter seems very straightforward, if you run it against another network without permission, it could be perceived as an offensive action against the site, and it could get you in a lot of trouble. From a security perspective, you should definitely run these steps against your own site, so you can better understand what information an attacker could gather. After you know this information, you will have a better idea of what things in your company need to be fixed and their priority.