CHAPTER 2 [:. FOOTPRINTING .:]

Now that you have gone through Chapter 1 — which could be somewhat boring to certain people — Chapter 2 is here to present you some exciting materials, intriguing your mind by walking you through some of the techniques involved in the very first stage of hacking. “Great, so is this where I can grab my keyboard and start 0wning a system?” Well, not so fast pal, you will not be able to hack anyone yet upon completion of this chapter. How do you expect to hack and “0wn” a system without knowing any relevant information about it, such as, its weaknesses, OS platform, or IP address? Therefore, this chapter introduces you to one of the most important aspects in a premeditated hacking process, *footprinting.* Footprinting is so essential and important that only a very few or no hackers can possibly succeed without first going through this stage. Having a good understanding of footprinting allows you to dig further into the art of information gathering. Later in the chapter, you will be introduced to methodology and tools that can be used to obtain general information about your target.

All the tools introduced in this chapter by no mean can represent for every single possible reconnaissance tools out there. They are being included here because of their popularity and usability in this genre. This chapter will walk you step-by-step through Module 2 of the official C|EH course outline. However, there are also some information included in this module that are considered as out-of-date and irrelevant to today’s world, so, there will be an additional section so-called “Bonus”, covering the latest or alternative security trends, tools, and techniques that can be used in lieu of the old ones.

There are also some changes between the layout of this chapter and the previous one. You should look out for the following icons, as they will be used throughout the whole book from now on:

🗡 Tools　　📜 Notes

🛡 Countermeasure　　🧰 Bonus Material

FOOTPRINTING

Because “footprinting” and “reconnaissance” are both related to each other in one way or another so before defining footprinting, let’s go back to the previous chapter for a quick review of what in the world is reconnaissance.

Reconnaissance is the preparatory phase of hacking in which the hackers will attempt to discover and gather as much information about the target as possible. All the information about the target is gathered to draw up a map or an outline of the target’s network infrastructure so that the hackers can have a better view about the environment which they are about to break into. Hackers who fail to obtain as much information about the target usually will find their tasks much more unpleasant than those who take time to investigate and understand the target environment inside out prior launching the attack.

Imagine how difficult it would be to hack into a network, blindly, without knowing a single bit about its associated features or inherent weaknesses. It is like trying to walk through a thick forest at night without having the handy map and torch, and of course, that is not always a good thing.

Footprinting is a passive and non-intrusive methodology of reconnaissance, allowing the hackers to collect all possible information about the target without the need of using aggressive reconnaissance techniques. Footprinting is the safest stage of hacking where the hackers can attempt to collect and harvest a handful amount of information about the target without giving the target any tips, or alerts, about those on going reconnaissance attempts. Information revealed during this footprinting stage can be very potential in aiding the hackers to complete subsequent phases in their hacking process. Some examples of which information the hackers can pull out from their footprinting include administrative contacts, technical contacts, network blocks, operating systems, domain name, IP address, and so on.

If you believe that such information is sacred and should be proper protected then you are right, but only to some extent. The information is not sacred; it is merely the information that is publicly accessible by anyone, anywhere, and at anytime. You can find it in the dumpsters around the target's office, in a particular public forum on the Internet, or even in those job vacancies posted on the Internet, and yet, without a sweat, you can just politely ask for such information too.

Virtually everyone can have access to that readily available information by spending a little bit of efforts and sifting through those huge public resources. The information is just there, freely available for anyone who wants to take it. Because of the fact that such information is too common and easy to obtain, not so many people can think of any reason why or how it can help one circumvent the security of their networks. The information indeed looks innocuous and harmless to those who own it, but once it is combined with a malicious mind, it can be the best tool to help one infiltrate into a network and it can be just as dangerous as any other hack tools or exploits out there.

The question that many people always ask is whether footprinting is necessary for a hack attack. The answer is, in fact, dependent on how you want to hack. There are always many different options and methods for you to hack and conduct an attack. You can consecutively try every single exploits and hack methods on this planet earth to hack a system without knowing anything about its weaknesses, architecture, or OS platform, and if you are lucky, you might be able to get in after all. However, trying to "0wn" a system that way will not only cost you a lot of work, time, and energy, but also, drastically reduce your success rate.

As an alternative, you may also choose to conduct a series of information gathering techniques to ascertain that you are on the right target and your exploits are relevant to the target's platform, so that you will not find yourself trying to spend years and years hacking a Linux server by using exploits coded for vulnerabilities of Windows' platform. Yet as you can see from several scenarios set out as above, a proper and premeditated hack always requires the perpetrator to have proper information about the target in place, by performing footprinting. Analyzing the information gathered from footprinting, the perpetrators, then, can make a decision and pick out the most appropriate methods and tools to hack into the system or network, which certainly can speed up the whole process as well as increase their chances of success.

In sum, footprinting is the pre-attack phase where the perpetrators not yet attack or do anything that would jeopardize the security of the target. Footprinting is a methodology encompassing non-intrusive reconnaissance techniques that allow the perpetrators to profile all potential aspects of the target prior launching the attack. No matter how much security measures employed at the target, chance for hackers to be detected and stopped at this stage is to be exceptionally minimal, if at all, because information about the target is publicly available and it is therefore cannot be protected. Footprinting is obviously necessary in any mature hack attack wherein it gives the hackers a unique profile about the target, such as, physical location, domain name, network range and remote access capabilities, as a result.

STEPS FOR GATHERING INFORMATION

Gathering information prior the attack is blatantly a major and yet time-consuming task that all ethical hackers like you must be able to accomplish. Information about the target can be enormous or minimal, depends on how the target choose to deal with "public" and "private" information. However, no matter how security paranoid the target can be, determined hackers will still be able to profile the target environment before executing the attack. It should be noted that there is no single rule or method for gathering information, hackers may use as many different reconnaissance methodologies as they wish to obtain information about the target from all possible sources. Figure 2-1 below illustrates three typical reconnaissance methodologies that cover all different types of target information that can be gathered.

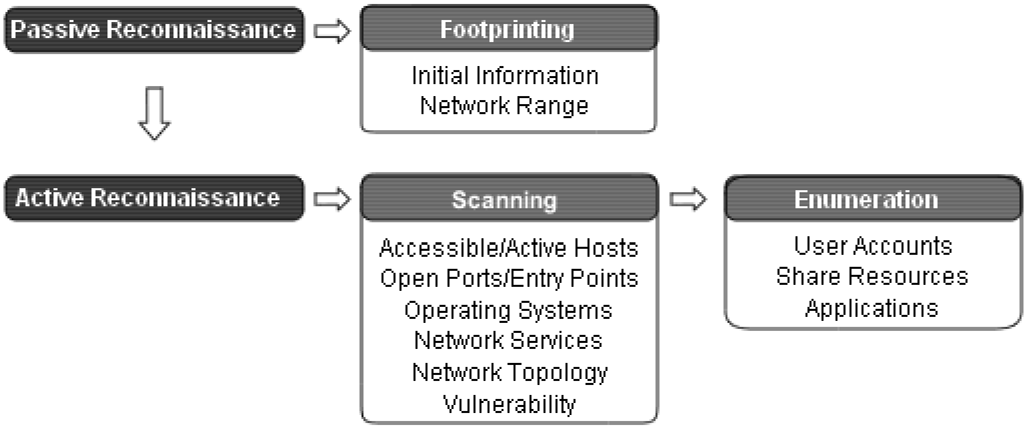


Figure 2-1. Information Gathering Methodology

Reconnaissance is the first phase of an attacker's process where everything about the target is gathered to feed into the information database, preparing for the Penetrating phase. Reconnaissance itself is divided into two subordinate phases, passive and active reconnaissance. Passive reconnaissance refers to the art of gathering information by using non-intrusive reconnaissance techniques, and if you say that sounds familiar then you are right, passive reconnaissance is also formally referred to as footprinting.

As in Figure 2-1, there are two kinds of information covered in footprinting stage, initial information and network range. Even though the information seems to be very trivial to obtain, its importance can't be stressed enough because it usually serves as the foundation for the attackers to construct a proper and serious attack against the target. Besides, obtaining such information alone in this footprinting stage is not enough to constitute an offense or lead the attackers to face legal consequences; therefore, it would be a mistake for not doing so.

> This chapter is dedicated to the discussion of footprinting only, which covers two types of information, general initial information and network range. Aggressive reconnaissance and network mapping techniques such as Scanning and Enumeration will be discussed in subsequent chapters, 3 and 4. They will cover complete facets about information gathering. For now, let's stick with footprinting and play the game "safe" first.

Active reconnaissance is the second reconnaissance method that helps the attacker discover more information about the victim, such as, active hosts on the network, entry points to the network, network services, and vulnerability. The difference between Scanning — the second phase in an attacker's process — and active reconnaissance, in my opinion, is too little to put them into different categories. Thus, I will use the term "Scanning" interchangeably to refer to active reconnaissance method. Some experts in the field seem to believe that Scanning comprises more aggressive probing and information gathering methods, and thus, it should stand alone as a single separate phase from reconnaissance. C|EH official course outlines also differentiate between Scanning and active reconnaissance, so just for neutralization sake, do not start a fight if you ever see someone challenging you this topic.

Another worth noting point is that enumeration normally resides in the Scanning phase of an attacker's process, as briefly explained in the previous chapter. However, due to the complexity and diversity of reconnaissance techniques involved in enumeration, discussions about enumeration will exclusively be dealt with in Chapter 4, instead of embracing them all into Chapter 3, which might make the chapter become intensive reading. Enumeration typically allows the attacker to uncover the following information: user accounts, share resources, and applications.

INITIAL INFORMATION

As the name implies, initial information is any information about the target that hackers can use as a starting point to construct the attack. Basic initial information of target environment, such as an IP address or domain name, is simply indispensable for a well-thought-out attack. How do you expect hackers to hack when they simply do not have any ideas of whereabouts or who their target might be? It is like trying to rob a bank without knowing the location or features of the bank in question. Hackers who spend a great deal of time to collect all possible information about the target, before launching the attack, are determined hackers going after a chosen, specific target and they are more likely to be successful than those who do not know anything about the target beforehand.

In some cases, hackers do not necessarily need to have initial information for a specific target when they just want to hack and "0wn" everything randomly. It might sound too extreme, but in reality, that is getting more common. Hackers nowadays just want to control as many machines as possible, so that they can use those compromised machines either to construct a larger and more sophisticated attack, or to merely mine for any valuable information randomly.

Initial information that can be gathered in this first stage include but not necessarily limited to the following:

- Physical location of the system
- Administrative/Technical Contacts (Mailing Address, Telephone, E-mail, etc.)
- Domain name and IP address of the system
- System architecture
- Financial Information

Initial information can be collected from various sources. The following are three typical information sources where hackers normally look out:

- Open source
- Whois
- Nslookup

The upcoming section provides you a list of techniques and tools that can be used to gather initial information from those information sources mentioned as above. Before moving on to the next section, I think a short discussion about the bad side and good side of using hacking tools, or exploits in general, is necessary here to enlighten those who are furiously making question like "Arg, using automated hacking tools, so are we all going to be script kiddies now?".

Tools are simply defined as any instruments that can be used to facilitate and relieve the burdens of manual operations. The definition also holds true regarding to hacking tools. Hacking tools were created to simplify the tasks of hackers and information system security professionals all around the world and the mere fact of using such tools cannot conclude or imply whether the actors are script kiddies or not. It all depends on which purposes the tools are being used for.

On the dark side, automated hacking tools are used by people who have limited knowledge about computer systems for the sole purposes of gaining unauthorized access to a system, and subsequently, destroying, stealing, or damaging the information stored on the system. This is where the script kiddies normally reside. Albeit the lack of knowledge and understanding of the attack mechanisms, script kiddies can give a careless to why, how, or by whom the tool was made, as long as the tool can help them gain access to the desired systems, they will use it.

On the brighter side, information system security professionals and ethical hackers, which including you, use the same hacking tools and methods as the script kiddies and malicious hackers to simulate a hack attack so that the result can be as close to the real situation as possible. Another reason behind that is to go “behind the scene” and understand the threats given by those exact same techniques and tools employed by the malicious hackers; and thereby, develop a better security solution to mitigate such risks. In fact, using hacking tools to serve for good and defensive purposes is highly encouraged in the real world, and that, certainly does not make one a script kiddy.

:.::.: OPEN SOURCE INFORMATION

> A few words to ease up those who are obsessed with the “open source” revolution, the term “open source” as mentioned in here has nothing to do with the little rebellious penguin or GNU, even though the intrinsic concept of “open source” is still applicable here. Hope that information gives you a peace of mind and makes you feel more comfortable and relaxing about the upcoming discussion of “open source information”. There is going to be no penguin, that’s a promise.

Open source information simply refers to any information that can be obtained by anyone without any restriction. Because such information is meant to be publicly accessible; accordingly, obtaining it turns out to be very easy and does not result in any legal implications. Many people also tend to think that once the information is public, it must no longer have any secret or value that might cause harm to or give other people advantages over the owner, and presume it is no big deal whether those public information is obtained or not. Yet the reasoning is only theoretically right. It all depends on who has a hold of such information, and how he or she is going to use it.

For example, from a posted job vacancy found in one of the job listing databases, a company advertises that they are in need of a network engineering who must be:

- Experience with knowledge of Microsoft Windows 2000, 2003 with domain controller, DNS Server, WINS Server, Active Directory and Group Policy
- Experience with knowledge of Mail Server Configuration with MS Exchange 2000, Exchange Server 2000/2003, Clustering System
- Experience with knowledge of SQL Server 2000
- Able to configure Terminal Server on Window 2003 Server
- Experience with knowledge of Inter-Networking Protocol, Network Security i.e. Firewall, IDS, Anti Virus, Internet Security, Proxy Server, Spam, Spy Ware
- Able to troubleshoot PC hardware and software and printers

The above information is obviously public information where everyone can obtain without a hitch. For those who are in need of a job, such information is invaluable in providing a scale showing expectations of the employer. However, hackers also find the information useful in helping them getting first hand knowledge about the target. From that job offer, it is safe bet to say that the target completely relies on Microsoft products to operate and run the show. The keyword “WINS Server” consolidates the guess and tips off the hackers that the target environment is Windows native. It is also trivial to note down that the target uses Microsoft Windows 2000/2003 as the main operating systems and has many different network services running. On top of that, it shouldn’t be much a problem to see that emails are routed and managed through Exchange Server 2000/2003, while computer data are stored and accessed through databases stored on database servers and operated by SQL Server 2000. More importantly, hackers now know the target network can be remotely accessed from the Internet through Windows 2003 Terminal Server. The second-last canon in that job vacancy post shows the target’s concerns about security issues, and that, alerts the hackers to put more thoughts into constructing the attack. Just from that tiny job opening advertisement, the company inadvertently gives out a lot of potential information and assists the hackers to carry out more intrusive information gathering attempts. For example, after going through the advertisement, the hackers should now be scanning the target network for port 1433 and 3879 to identify the database and terminal servers, respectively.

Another place where hackers can look for open source information is, simple enough, the target’s website. Merely visiting and browsing through related pages of the site can help the hackers elicit tremendous information about the target, such as, the platform which the web server is currently running on, the web server software, or the scripting language used if the site is serving dynamic pages. The hackers can also view HTML source code of the pages to seek for more information including hidden comment tags and directory structure. Optionally, hackers may use offline browsing utilities such as Teleport Pro, HTTrack, Wget to download and mirror the entire site as well as directory structures of the site for offline viewing and auditing. Teleport Pro runs on all Windows platform and requires users to pay a small fee for the license, while HTTrack and Wget are both freeware which will run on all platform, including UNIX, Linux and its variants.

After visiting the site, the hackers should have had at least some initial, basic information pertaining to the target. Now the next step is to use such information to consult with their “friend” to seek for more information. Neither their friend is an insider or expert in hacking. Their friend is a mere search engine. “Google is your best friend”, that popular saying in the hacker’s community is indisputable where the search engine is getting much more powerful comparing to their first arrival in 1997. Google now claims their search engine is capable of searching over 8 billions of web pages, which is an enormous amount of resources. By feeding the target organization’s name or domain as the keyword to search for, the returned results can yield a lot of potential information related to the target, including abandoned web sites at the target, past events, target’s partner, press releases and related articles. Another way to quickly identify partners of the target is to use major search engines like AltaVista to search for any site that has link referring to the target website. In doing so, hackers might also be able to uncover any vulnerable sites that were accidentally setup and left opened by the target administrators.

For more phenomenal results, hackers normally conduct more sophisticated search queries with the use of Boolean expressions to collect more sensitive information, such as private customer’s login page, customer lists, administrator’s name, password, and so forth. It never ceased to amaze me how such confidential and sensitive information can be easily searched for and obtained through some popular search engines. Recently, there has been a case where a hacker was able to get numerous password files stored on many MIT servers merely by using Google and its associated features. Feel free to download and read the unofficial Phrack, Issue 63 available at http://www.phrack.nl for more information regarding the incidents. Just don’t take everything you read in there for granted!

If results returned from one search engine are not enough, then there are also many other search tools, either web-based or program-based that display search results from multiple search engines. Dogpile.com, FerretPro and Copernic Agent are all capable of submitting search queries to multiple search engines, and display the results through a single portal or interface. For a free and popular web-based version, Dogpile.com seems to be the best choice though it lacks of some advanced features implemented in some commercial search tools. FerretPro or Copernic Agent Personal or Professional are both commercial program-based search tools that not only do allow you to search multiple search engines, but also make it possible for you to search and retrieve information from many other sources, including USENET, IRC, newsgroup, file databases, and so on.

FerretPro and Copernic Agent can be found at http://www.ferretsoft.com and http://www.copernic.com respectively

USENET or newsgroup is where people can join to discuss or seek for quick help and solutions for problems under the same subject matter and it is yet another popular information source. What do you do when you need other people to help you fix a problem related to your newly bought CISCO router? Of course, you have to provide as much information pertaining to the problem as possible in order to receive the most appropriate solution or answer. This is exactly where the problem lies. Hackers can always filter for any posts come from the target organization’s domain and sift through those posts describing problems or seeking for help to gain some basic knowledge about the infrastructure of the target system or network.

Finally, to gain some more essential inside information about a company such as annual reports, filings, financial related news, and turnover or merging status, hackers normally pay a visit to http://finance.yahoo.com and the SEC (Securities and Exchange Commission) Edgar database at http://www.sec.gov. It is worth noting that the Edgar database is only capable of providing information pertaining to US companies. For companies located in the UK, hackers can always visit http://www.companieshouse.gov.uk to get the information. Information about the merging status between two companies is the information that hackers and opportunists would normally love to possess. When one company is in the process of merging with another company, security measures are often reconfigured to be more relaxed at both sides so that the mergence can happen smoothly, without any hindrance. Such bad practice inadvertently creates a security loophole where the hackers can jump in and enjoy the advantages of “relaxed” security measures.

COUNTERMEASURE: MINIMIZE THE EXPOSURE

Organizations are actually giving out a lot of seem-to-be-harmless information about themselves without pausing and giving it a second thought. Even though it is true that public information can cause no instant loss or harm and can be very valuable to people like the clients or investors; however, it is imperative that such information must also be controlled and checked on a regular basis, so that they do not inadvertently provide the attackers a detailed roadmap to the network.

Strictly speaking, public information is certainly something that you can't have full control. Some of the open source information mentioned above must be made publicly accessible, and it is just the fact that you can only do a little or nothing about it. Yet something that you can do to prevent leakage of sensitive information is not to leak it at the first place. Disseminating the wrong kind of information is just as bad as proactively opening up opportunities and inviting hackers to break into your network. Thus, you have to be aware of which information should be made public and which should be not, and the only way to do that is to classify your information in compliance with your security policy. Avoiding unnecessary information and comments from your web pages is also one of a very few ways that would help you not to be an attractive target.

:.::.: WHOIS

Every connected computer on the Internet is assigned a unique address, or also known as an IP (Internet Protocol) address, allowing a computer to locate and communicate with one another easily. The Internet Protocol address space, Autonomous System (AS) numbers, "in-addr.arpa" (inverse mapping), and other Internet numbering resources are distributed, allocated, and managed by four Regional Internet Registries (RIRs) in the world, which represent for four major geographical regions. Thus, the RIRs are usually where hackers should first be looking at to find out more information pertaining to a particular registered resource, such as, contact details of the organization or owner of a block of IP address. Exhibit 2-1 lists the four primary RIRs and their respective regions.

RIR	Region	Web Site
APNIC	Asia Pacific	http://www.apnic.net
ARIN	Northern America and sub-Saharan Africa	http://www.arin.net
LACNIC	Latin America and the Caribbean	http://www.lacnic.net
RIPE NCC	Europe and Middle East	http://www.ripe.net

Exhibit 2-1. The Four Primary RIRs (Regional Internet Registries)

An IPv4 address is a 32-bit binary numbers and normally written down as four decimal numbers separated by periods, so that the address can appear to be more human readable. Even so, users tend not to be really good at remembering a whole bunch of numbers; so some really smart geeks decided to come up with a new and easier to memorize addressing scheme, which is what you often see nowadays, *Domain Name System*

(DNS). DNS allows Internet users to conveniently refer or locate a computer by using a pronounceable name — domain name — in lieu of the complicated numerical IP address.

Anyone who wants to register a domain name, for instance, www.gotrice.com, must somehow at least provide some personal, geographical, or contact information to the domain name providers, or also called *registrars*, and those registrars subsequently submit to and store the provided information in a central database so-called the *registry* or whois database. The information is crucial for the registrar to verify the owner or *registrant* of the domain name, in the case where the registrant decide to delegate the domain name registration to another registrar or person. Moreover, the information are made publicly accessible for the purposes of enforcing the trademark and intellectual property laws, facilitating the law enforcement to control illegal activities on the Internet, and of course, allowing the domain name holder to be reached quickly in case of emergencies or technical problems. Therefore, providing the registrar with the required and necessary information is more likely to be an obligation of the domain name owner, rather than just an option.

Information stored in the whois database can be dig up by using a whois client. Whois client is conveniently built-in and available in almost every UNIX and Linux platform. Normally, you can just type `whois` at the command prompt to start using it or `man whois` for more information on its usage and options that you can muck around with. If you are not a big fan of console-based utilities, you might find the Xwhois client nifty. Xwhois client does what its name implies; making it possible for UNIX user to carry out whois queries through X interface or GTK+ GUI toolkit, Xwhois client can be downloaded from http://c64.org/~nr/xwhois/. Yet Windows users are not left behind either, there are many freeware and shareware network utilities that come with whois client allowing Windows users to do what UNIX users can normally do with `whois`. Whois client is also available in CGI form and you can easily submit whois queries via web interface instead of downloading and installing the program. In general, functions and switches found in those whois clients should be identical, or the same for most parts, and the results returned from your queries should also be the same. Exhibit 2-2 lists some of the popular whois clients for Windows users and places to download the clients.

Program-based	**Web Site**
Sam Spade (Freeware)	http://www.samspade.org/ssw/
Netscan Tools (Shareware)	http://www.netscantools.com
SmartWhois (Shareware)	http://www.tamos.com
GTWhois (Freeware)	http://www.geektools.com
Saeven (command-line whois)	http://www.saeven.com/sware
Web-based (CGI Interface)	
Sam Spade	http://www.samspade.org
Network Solutions	http://www.networksolutions.com
AllWhois	http://www.allwhois.com
Arin	http://www.arin.net

Exhibit 2-2. Whois Clients for Windows or UNIX platform

Up to now, you should know which entity is responsible for distributing and managing IP address space, what sort of information is stored in the whois database, and where to get whois clients. It is now time to look deeper into various different whois queries types that hackers frequently use to gather information about the target organization.

Knowing the IP address or domain name of the target network before executing the attack is certainly one of the most important things that all hackers must be well aware of, since that is the only way that can help them locate the target on the Internet. The domain name or IP address of the target can usually be extracted from open source information, but if it was such a bad day for the hackers and nothing about the target could be really found from open source information, then they would have to rely on different types of whois queries to footprint the target network.

Since the IP address or domain name of the target is not yet known up until this stage, the first thing in the to-do-list should be to search for any potential domains that match to the inputted keyword, in this case, name or business type of the target organization. For example, if “gotrice” is the name of the target organization and it is only thing that you know about, then, searching for any domain that has the word “gotrice” or “rice” should be a good starting point in identifying the address or domain name of the target. As you will see with the following example, the query is submitted by using the built-in whois client available on almost every UNIX platform — `whois`. There are two different ways to make `whois` works, but regardless of which one you use, your query will still receive the same result. Pick the one that you like best and stick with it. The book will use the first method, as listed below, since it seems to be unambiguous and easier to interpret.

```
[bash]$ whois -h whois.crsnic.net gotrice.
[Querying whois.crsnic.net]
[Whois Server Version 1.3]

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.

GOTRICETOGO.COM
GOTRICER.COM
GOTRICEPRODUCTIONS.NET
GOTRICEPUNK.COM
GOTRICEDUDE.COM
GOTRICE4U.COM
GOTRICE.NET
GOTRICE.COM
```

This is the second method to get the thing done:

```
[bash]$ whois "gotrice."@whois.crsnic.net
[Querying whois.crsnic.net]
[Whois Server Version 1.3]

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.

GOTRICETOGO.COM
GOTRICER.COM
```

```
GOTRICEPRODUCTIONS.NET
GOTRICEPUNK.COM
GOTRICEDUDE.COM
GOTRICE4U.COM
GOTRICE.NET
GOTRICE.COM
```

Note how the whois query included the "." behind the keyword "gotrice" and used the whois database at whois.crsnic.net server. The "." is a wildcard character which was used in the query to ask the whois server — crsnic.net — to search for any domain that starts with the word "gotrice" instead of just searching for exact matches of "gotrice". As a result, domain names like `gotricepunk.com` and `gotriceproductions.net` were also included in the returned result. The `-h` followed by the address of the whois server (without the leading http://) in the first query is to set the query using the whois database of the specified host for information. Another way to specify an alternate whois database is to use the `@` option as in the second query. It is worth noting that despite the different uses of options, both of the queries still yielded the same results. Therefore, choosing the one that suit you best is just a matter of personal style, though using the first method is usually more favorable. Whois database at whois.crsnic.net should be used here because you are still in the stage of mining for any potential domain name belonging to the target, as well as its associated registrar.

If gotrice.com sounds too convincing and you have decided to find out more information about the domain name to ascertain if it indeed belongs to the target, you need to first find out its associated registrar — domain name provider — before you can actually perform detailed whois query on that specific domain name.

Before the revolution of anti-monopoly in managing, providing, and distributing domain names, conducting whois queries was so much easier because Network Solutions (http://www.networksolutions.com) was the only place where you ever need to go to perform whois query on any domain name ending with .com, .net, .org, .edu, and few others. Now with many other accredited registrars available, it is your task to pick out the most appropriate or closely matched domain name to your target organization, and subsequently, identify the associated registrar with that particular domain name in order to lay your hands on to the detailed information. Identifying the registrar of a domain name is not that much difficult, the following whois query will help you complete the task just fine:

```
[bash]$ whois -n -h whois.crsnic.net gotrice.com
[Querying whois.crsnic.net]
[whois.crsnic.net]

Whois Server Version 1.3

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

   Domain Name: GOTRICE.COM
   Registrar: GO DADDY SOFTWARE, INC.
   Whois Server: whois.godaddy.com
   Referral URL: http://registrar.godaddy.com
   Name Server: SHADY.SHADOW.COM
   Name Server: WUTANG.CLAN.ORG
   Name Server: NO.SHADOW.COM
   Status: ACTIVE
   Updated Date: 16-jun-2004
   Creation Date: 22-jun-1997
   Expiration Date: 21-jun-2006
```

Voila! The whois query was successfully executed although its syntax was just a bit different from the first one. The `–n` option was used to disable the redirection feature that transfers the query from one whois server to another automatically. If `–n` wasn't used in the query, `whois` would transfer and submit the query to the appropriate whois server of the domain gotrice.com, saving you from manually submitting another whois query. The –n option was used only for the purposes of showing you how to specifically gather information about the associated registrar of a domain name should is there a need to do so. Nonetheless, in the real world, you should of course choose the most economical way to carry out a whois query, which is avoiding the use of `–n` unless you have some specific need for it.

According to the output from the query, it is easy for you to see that the target is with the Go Daddy Software, Inc. registrar and information about the associated whois servers and name servers of the domain gotrice.com was also uncovered in addition to the registrar information. From the two previous queries, you should have all the necessary information ready and it is now time to pay the whois server at Go Daddy Software a little visit to get more detailed information.

In the real world, hackers normally can jump straight to perform whois query on a domain name, skipping the need of searching for potential domain name relating to the target organization, since it's relatively easy to extract such information from open source information. However, as you have seen earlier, even if the hackers were not able to get anything useful like domain name of the target network from open source information, they might still be able to get a hold of it by doing their "magic" with whois queries.

Whois query type that searches for the associated registrar and whois server of a particular domain name is called the **Registrar** query type.

Given the information provided from your last whois query, you should now be specifying `whois` to use the whois server of the registrar Go Daddy Software, Inc., which is whois.godaddy.com, in order to narrow down to more specific information related to the domain gotrice.com. Remember, CRSNic.net whois server is not responsible for providing specific information relating to a particular domain, its tasks are merely to search for .com, .net, and .edu domain names and provide information about the associated registrars. Only the registrar of the respective domain name has the ability to provide you the details, including the registrant information, primary and secondary authoritative name servers, and contact details.

```
[bash]$ whois –h whois.godaddy.com gotrice.com
[Querying whois.godaddy.com]
[whois.godaddy.com]
The data contained in Go Daddy Software, Inc...

<messages truncated for brevity>
```

```
   Registrant: Andre Volwanky
      777 Walk A Way
      San Jose, California 95121
      United States

      Registered through: GoDaddy.com
      Domain Name: GOTRICE.COM
         Created on: 22-Jun-97
         Expires on: 21-Jun-06
         Last Updated on: 16-Jun-04

      Administrative Contact:
         Andre Volwanky Dr.Dre@gotrice.com
         777 Walk A Way
         San Jose, California 95121
         United States
         (408) 777-7777      Fax --
      Technical Contact:
         Andre Volwanky Dr.Dre@gotnorice.org
         777 Walk A Way
         San Jose, California 95121
         United States
         (408) 777-7777      Fax --

      Domain servers in listed order:
         WUTANG.CLAN.ORG
         SHADY.SHADOW.COM
         NO.SHADOW.COM
```

The whois query result is pretty obvious and self-explanatory, providing all essential information related to the domain “gotrice.com”. First off, the registrant information is shown at the top of the output, showing the person whose name is Andre Volwanky with the associated physical mailing address is the registrant of the domain name. If you know that your target organization is named gotrice and located somewhere in San Jose and the registrant of the domain name gotrice.com is also happened to be located somewhere in San Jose, then it is safe bet for you to say that gotrice.com must be belonged to or at least have something to do with the target. Even so, keep in mind that postal mailing address shown in the registrant field of the output does not always necessarily reveal the real address or location of the organization. Followed by the registrant information are the creation and expiration dates of domain name as well as the last modification date of the whois records. Then, administrative and technical contact details of the domain name gotrice.com are shown, which reveal phone number, email address and mailing address of Mr. Andre Volwanky. The last part of the whois output lists the primary and secondary authoritative name servers of the domain. Mind you that all information and references shown in the output have been changed to be fictitious to prevent improper uses of such information.

All in all, specifying `whois` to use the whois database of the corresponding whois server of the domain name normally helps you uncover the following information:

- Registrant name
- Physical location of the target (this may not always be accurate)
- Administrative and technical contact details

- Creation and expiration date of the domain name
- Last modification of the whois records
- Primary and secondary name servers

It might come to your attention why anyone would want to get such detailed information since it would not provide him or her instant access to the desired network. For those hackers who are superficial and easy to get discouraged, whois records of a domain name are only as good as high scores records of Pinball players, and apparently, they appear to be useless. Contrarily, visionary and determined hackers find the information invaluable in providing them with useful information about the organization that they can rely on to conduct social engineering attack, which is the art of deceiving innocent and naïve people into giving out sensitive information. In addition, the phone and fax numbers given away by the query result also facilitate the hackers in carrying out war-dialing, which is a hacking technique that allows the hackers to infiltrate into a network by consecutively dialing into a range of defined phone numbers to look for PBX or any system left connected to a modem through the telephone line.

Yet life is not all beer and skittles, as you will certainly encounter the times when you are not able to retrieve all the information related to a particular domain name. That usually happens when you perform whois query against a corporate domain name, since corporations will always try to provide limited or generic information about their collective network to decrease the likelihood of network abuses. Let's specify `whois` to use the domain yahoo.com instead of gotrice.com to see if the result would be any different between a corporate and a private or personal domain name.

```
[bash]$ whois -h whois.alldomains.com yahoo.com
[Querying whois.alldomains.com]
[whois.alldomains.com]
Alldomains.com - The Leader in Corporate Domain Management
...<messages truncated for brevity>

Registrant:
        Yahoo! Inc. (DOM-272993)
        701 First Avenue
        Sunnyvale CA 94089
        US

    Domain Name: yahoo.com

        Registrar Name: Alldomains.com
        Registrar Whois: whois.alldomains.com
        Registrar Homepage: http://www.alldomains.com

    Administrative Contact:
        Domain Administrator (NIC-1382062)  Yahoo! Inc.
        701 First Avenue
        Sunnyvale CA 94089
        US
        domainadmin@yahoo-inc.com
        +1.4087654321
        Fax- +1.4087654322
    Technical Contact, Zone Contact:
        Domain Administrator (NIC-1372925)  Yahoo! Inc.
        701 First Avenue
        Sunnyvale CA 94089
```

```
          US
          domainadmin@yahoo-inc.com
          +1.4087654321
          Fax- +1.4087654322

     Created on..............: 1995-Jan-18.
     Expires on..............: 2012-Jan-19.
     Record last updated on..: 2004-Nov-24 16:01:39.

     Domain servers in listed order:

     NS4.YAHOO.COM               63.250.206.138
     NS5.YAHOO.COM               216.109.116.17
     NS1.YAHOO.COM               66.218.71.63
     NS2.YAHOO.COM               66.163.169.170
     NS3.YAHOO.COM               217.12.4.104
```

You can clearly see that this whois query and the previous one provided the same amount of information but with two slightly different contents. With the first whois query for the domain name gotrice.com, a more specific type of information about the domain name was revealed, such as the registrant's name, phone numbers, physical mailing address, e-mail address, and administrative and technical contact details. However, comparing the information provided from the first query to the second one, you can see the second whois query result only showed a generic descriptive role-based name and contact information provided by Yahoo! Inc., instead of a specific name or contact details of the person who is in charge of managing and updating the domain name. Imagine which one of the following information is better for hackers to construct social engineering attack with greater effects, a specific name and address of the responsible personnel like "Andre Volwanky" as in the first query? Or a generic descriptive role-based name "Domain Admin" and its associated address as in the later one? Of course, the information provided in the first query will always be more helpful to the hackers and their wicked "social engineering" attack since it provides the hackers with more detailed information.

Domain query allows you to search for all information related to a particular domain name and you need to know the associated registrar of a domain before being able to submit this type of whois query. However, most of the whois clients you see these days are capable of automatically redirecting your whois query to the most appropriate or associated registrar and whois server of the domain name, which also means you no longer have to specifically perform a Registrar whois query before being able to submit a whois query on a domain name.

So far, you have been shown how to gather all possible information related to a specific domain name, but what if you just want to perform a broader query to obtain registration details of a particular network block, or an AS (Autonomous System) number, or an IP address? Given that the four RIRs are responsible for allocating, distributing, and managing the Internet address resources, including the Internet Protocol address space v4 and v6 and Autonomous System numbers; it is obvious that your queries must consult with whois servers of the four RIRs in order to obtain such information.

Besides providing registration details pertaining to a particular network block, the RIRs' whois database also contain information about network blocks that are allocated for and owned by an organization. In the following example, the query uses whois server of the American Registry for Internet Numbers (ARIN) to determine all the networks owned by Yahoo! corporation. It is worth noting that the query must consult the ARIN's whois database in order to obtain any information related to Yahoo! since the corporation belongs to the Northern America region.

```
[bash]$ whois -h whois.arin.net Yahoo!
[Querying whois.arin.net]
[whois.arin.net]
Yahoo! (YAOO)
Yahoo! Broadcast Services, Inc. (YAHO)
Yahoo! Broadcast Services, Inc. (YBS-2)
Yahoo! Inc. (YAHOOI-2)
Yahoo! (AS10310) YAHOO-1    10310
Yahoo! (AS26085) YAHOO-2    26085
Yahoo! (AS26101) YAHOO-3    26101
Yahoo! (AS32116) YAHOO-4    32116
Yahoo! Broadcast Services, Inc. (AS5779) Y-BR-SERV    5779
Yahoo! A-YAHOO-US2 (NET-216-115-96-0-1) 216.115.96.0 - 216.115.111.255
Yahoo! A-YAHOO-U23 (NET-66-218-64-0-1) 66.218.64.0 - 66.218.95.255
Yahoo! LEVEL3-YAHOO-1 (NET-67-28-112-0-1) 67.28.112.0 - 67.28.115.255
Yahoo! YAHOO-3 (NET-66-94-224-0-1) 66.94.224.0 - 66.94.239.255

...<results truncated for brevity>

# ARIN WHOIS database, last updated 2004-11-25 19:10
# Enter ? for additional hints on searching ARIN's WHOIS database.
```

The ARIN's whois server replied to the query with a long list containing information about network blocks that are owned by Yahoo!. To further examine and get more specific information related a particular network block; you can submit another whois query to ARIN as in the following:

```
[bash]$ whois -h whois.arin.net 66.94.224.0
[Querying whois.arin.net]
[whois.arin.net]

OrgName:    Yahoo!
OrgID:      YAOO
Address:    701 First Avenue
City:       Sunnyvale
StateProv:  CA
PostalCode: 94089
Country:    US

NetRange:   66.94.224.0 - 66.94.239.255
CIDR:       66.94.224.0/20
NetName:    YAHOO-3
NetHandle:  NET-66-94-224-0-1
Parent:     NET-66-0-0-0-0
NetType:    Direct Allocation
NameServer: NS1.YAHOO.COM
NameServer: NS2.YAHOO.COM
Comment:
RegDate:    2003-07-17
Updated:    2003-07-17

OrgTechHandle: NA258-ARIN
OrgTechName:   Netblock Admin
OrgTechPhone:  +1-408-765-4321
```

```
OrgTechEmail:  netblockadmin@yahoo-inc.com
```

The result returned from the whois query is rather self-explanatory. Some of the typical ones include mailing address of the network block's owner —Yahoo! — provided at the top part of the whois result followed by essential information related to the network block, such as, network range, name servers, network handle, network name, and so on. In the last part, technical contact details of the associated network block are listed so that the responsible personnel or department managing the network block can be contacted in the event of problems.

> To obtain all information related to a particular network block or IP address, you need to submit a **Network** whois query to one of the four respective RIRs whois database for such information.

In addition to submitting query to the ARIN's whois database to mine for any network blocks belong to Yahoo! Inc., you can also submit a whois query that searches for all email addresses belong to Yahoo! Inc. as in the following example:

```
[bash]$ whois -h whois.arin.net @yahoo-inc.com
[Querying whois.arin.net]
[whois.arin.net]
Michael, Doug (MD11-ARIN) netblockadmin@yahoo-inc.com +1-408-009-1122
Kenny, Krewlio (AKE8-ARIN) kekrew@yahoo-inc.com +1-408-345-6789
Free, Azyuky (AZ861-ARIN) freeazyuky@yahoo-inc.com +1-619-191-9191
Welly, Hogger (WHS1056-ARIN) wellhog@yahoo-inc.com +1-212-200-0022
Andre, Pucket (AT394-ARIN) domainadmin@yahoo-inc.com +1-408-000-1100

...<results truncated for brevity>
```

Note that the all the whois query results provided throughout this chapter were deliberately changed to protect the host or network in question. The string shown in parenthesis following the name field is the database handle. A handle represents for or points to a specific and unique record in the whois database, which means you can easily access to the record anytime by submitting a whois query on the relevant handle.

```
[bash]$ whois -h whois.arin.net AKE8-ARIN
[Querying whois.arin.net]
[whois.arin.net]

Name:        Kenny, Krewlio
Handle:      AKE8-ARIN
Company:     Yahoo Financial
Address:     12345 Nokes Boulevard
City:        Sterling
StateProv:   VA
PostalCode:  20166
Country:     US
Comment:
RegDate:     2004-03-11
Updated:     2004-03-11
Phone:       +1-408-345-6789 (Office)
Email:       kekrew@yahoo-inc.com
```

> Searching for all information related to a person by using his or her associated

database handle is called a **Point of Contact** (**POC**) whois query.

Before hacking and spamming becoming as ubiquitous as today, registrars used to have a relaxed policy upon whois queries, and practically, there was no restriction regarding what type of whois query one may submit. Increasingly, Network Solutions and many other major registrars have realized that information stored in the whois database is extremely valuable to spammers, opportunists, and hackers; as a result, they have decided to disallow a certain type of whois query that exposes the clients to greater threats by giving away too much information. One of which, in the disallow list, is the **Organizational** whois query that allows one to seek for all information related to a specific organization, including all the associated domains of the target organization. Exposing this kind of information to the wrong crowd like hackers is relatively dangerous, since it provides the hackers with more options and entry points to attack the organization.

COUNTERMEASURE: MINIMIZE EXPOSURE OF PUBLIC WHOIS RECORDS

As you folks can see, with just a little bit of effort and of course several whois queries, a lot of detailed information about an organizational domain name or network can be easily obtained, including the registrant's name, phone and fax numbers, electronic and physical mailing address, administrative contact details, and so forth. Such information must also be promptly submitted to the associated registrar — at the same time when the domain registered on the Internet — for legitimate uses. Furthermore, the registrant or domain name owner will not be able to control who or which entity is allowed to have access to the information, since it is rather an obligation of the registrant to make it publicly available. Yet despite the fact that the information itself has several legitimate uses from the registrar perspective, it is undeniable that the information can also be used to serve for illegitimate purposes, by the hackers, hijackers, spammers, identity thieves, or anyone who has the intention to harm the network.

Consequently, organization should try to avoid providing or giving away too much specific information about the domain name or the collective network. One way to achieve that is to use a generic descriptive role-based name and contact address so that the hackers and opportunists won't be able to use such information against the network. If information related to the domain name must be provided at a specific or detailed level in some ways, then, it is incumbent upon the organization to alert and tell the key staff and users to treat such information as public information, not to give them a higher priority, and apply the same security procedures should the information is offered by anyone.

Information regarding phone and fax numbers of the organization must also be given great consideration because hackers often use it to deduce the phone number range of the organization to carry out war-dialing or social engineering attack more effectively. A toll free number like 1-800 or a generic phone number that is dedicated for reporting problems, or anything related to the organizational domain or network should be used whenever possible.

Seeing the forthcoming revolution of hacking and how whois records could have greatly facilitated the hackers' tasks, many accredited domain name registrars also come up with a privacy protection scheme to limit the exposure of personal or detailed information when a domain name is registered. For example, Network Solutions is offering a service known as "Private Registration" which allows the domain name holder using an alternate phone number, postal address, and email address assigned by Network Solutions as the registration information for the domain name. Detailed personal information pertaining to the registrant will then be privately kept and held by the Network Solutions registrar for verification purposes, should any request, delegation, or problem related to the domain name arise. Note that not only do Network Solutions remove sensitive information from public disclosure, but all domain-related data are also removed from the Bulk whois list, which is a list containing third parties that are given full access to the registrar's whois data for an annual fee.

Yet hackers using publicly exposed whois information to further hack and make their way into the network is not the only thing there is for the registrant to worry about. Domain name holder also has to face another common irritating issue in today's world, *domain hijacking*. Domain hijacking refers to the act in which the hijackers or hackers will attempt to steal the target organization's domain name instead of hacking into the network, which can be much more difficult wherein convincing the registrar with the submission of a fraudulent domain transfer request is all it takes to hijack. Another popular method to hijack a domain name is to provide the authoritative DNS server of the domain with false information. Reasons for the hijackers to hijack a domain can vary, but most of the times, domain hijackers use the hijacked domain to blackmail the registrant or to redirect all the clients to explicit website providing false services with the intent to defraud. Protecting the domain name from being hijacked is more likely to be a mutual responsibility of the two parties, the registrant and the registrar. Any registrant or entity that is fully aware about security issues, or paranoid about being hijacked, should only pick a registrar that implements proper verification procedures and strong authentication scheme before accepting a transfer or modification request related to the domain name. Additionally, the registrant should look for any registrar providing the "registrar lock" service, which is a service that locks all information in a domain record at the registry level to prevent the domain from being unauthorized transferred, deleted, or modified by a third party.

:.::.: NSLOOKUP

Nslookup is a neat network utility that comes with Linux, UNIX, and Windows operating system by default, which you can use to diagnose and debug any problems associated with your Domain Name System (DNS) server, as well as to query the DNS server for any information related to a remote host. Before diving into the details and exploring how nslookup really works, it is imperative that you should have some fundamental knowledge about the underlying mechanisms of the DNS.

As briefly introduced earlier in Whois section, the development of DNS is to allow users to use a pronounceable name — host name — instead of a complicated string of numbers, to locate a computer on a network. However, attempting to identify and locate a unique computer on an enormous and intricate network like the Internet by using only its associated host name is more like looking for a needle in a haystack. The computer needs to have a *domain* attached onto its host name in order to be worldwide reachable by Internet users; and the fine combination between a host name and a domain gives you what formally known as *Fully Qualified Domain Name* (FQDN). For example, a FQDN like dephunk.ecqurity.com will help the name servers unambiguously locate the computer with the host name "dephunk" in the "ecqurity.com" domain.

As defined in RFC (Request for Comments) 1034, a DNS consists of three different components. They are:

- *Name servers* — Maintaining cache and containing a portion of a domain database or namespace in what is called a *zone*. Name servers are also responsible for making the information available to the name resolvers by either providing answers, or delegating name resolution queries to the appropriate name servers. A single name server may have authority for many zones; and on the same hand, one zone may also have more than one name server to improve load balancing and ensure redundancy.
- *Domain database* or *namespace* — Storing information related to a domain including IP addresses, *mail exchange* records, aliases or *canonical names* of the hosts. Information about other name servers is also stored here. This is exactly the kind of information that *nslookup* and *you* will be very much interested. Records stored in a domain namespace will be discussed shortly.
- *Name resolvers* — Responsible for making connection to the name servers to look for information stored in the domain namespace or database. The "name resolvers" are normally just library functions integrated into almost every Internet application on the users' computers, or any application that has the need to lookup addresses. Alternatively, name resolvers are also available in program form, such as nslookup and dig, allowing you to connect to and interact with the desired name servers instead of merely relying on the "gethostbyname" or "gethostbyaddr" library function.

Users usually interpret or type a domain name from right to left; but contrarily, DNS prefers to work with a domain name in a backward direction — left to right — and in a hierarchical approach. Figure 2-3 illustrates an example of the hierarchical structure of the DNS database.

From the illustration shown above, you can see the hierarchical structure of the DNS database is very much resembling to the tree structure of the UNIX file system where you must travel from the root node through any subordinate nodes in the hierarchy to get to the desired or specific node.

At the top of the DNS hierarchy is a single *root* node or domain represented by a period "." but normally you will only see it as a node with null label. Underneath the root node are the *top-level domains* (*TLD*s) which provide general description about the owner of second-level domains; some famous examples include .com, .net, .edu, .org, and .gov.

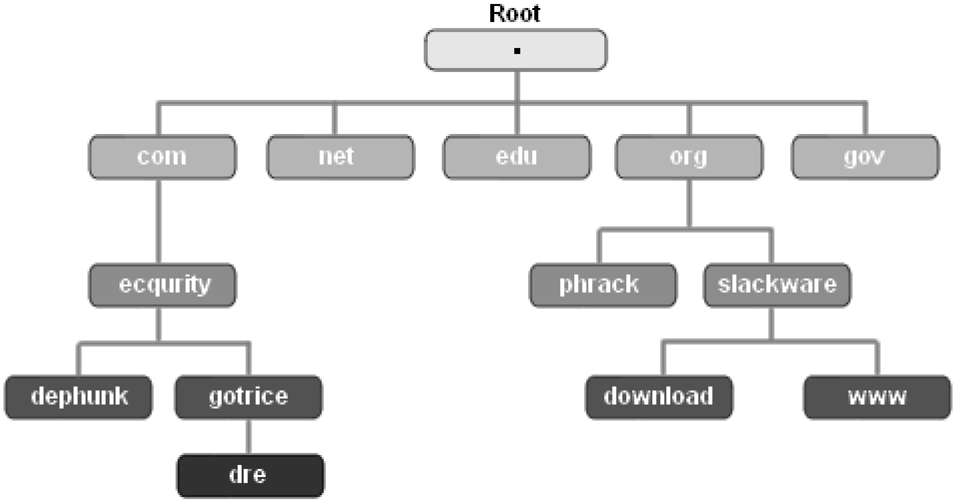


Figure 2-2 . The hierarchical structure of the DNS

Below the TLDs are the *second-level domains* where things start getting brighter for the users, since now they have the authority to register and give the second-level domains some names or labels. However, each node in the second-level domains must not have a null label — which is exclusively reserved for the root node only — and its label must be unique, that is, no two nodes in the same level domain may have identical label or name. For example, if the second-level domain of the domain name `phrack.org` is `phrack` and `.org` is its TLD — as can be seen in Figure 2-3 — then under the `.org` TLD, no one may register and use `phrack` as a label for any other second-level domain, assuming the term “domain hijacking” is not in his or her dictionary.

An entity who has authority over a node in the second-level domains — the registrant — may create as many additional levels of *subdomains* as they wish without having to rely on or ask for permission from the registrar — domain name provider. For this reason, beneath the second-level domains, depending on various needs of the registrant, there may be many subdomains or no subdomain at all between the host name and the second-level domains. As you can see in Figure 2-3, only the two nodes named `dephunk` and `dre` are actual host names which give the name servers precise locations of those specific computers in the `ecqurity.com` domain. The node `gotrice` is just a subdomain in the second-level domain `ecqurity`, despite the fact that `dephunk` and `gotrice` are both in the third-level domains or same level in the hierarchy.

In all, the hierarchy of a domain namespace theoretically can have up to 127 levels in depth, a maximum of 63 characters per label, and the overall length of a domain must not exceed 255 characters. Nevertheless, having a lengthy, complicated, and “provocative” domain name like that would for sure cause a lot of irritation and deter users from visiting your web site, regardless of how beautiful or great it might be. So be wise when choosing a domain name.

As explained in previous section, name servers contain a portion of a domain database or namespace in what is called a zone. A zone subsequently stores all information, or *resource records*, associated with a particular domain into a zone file; and it is where the name servers can refer to anytime they need to respond to queries or requests from the name resolvers. Resource records responded by the name servers should have the following fields:

- *Domain Name* — Identifying the domain name or owner of the records
- *Record Types* — Specifying the type of data in the resource record
- *Record Class* — Identifying a class of network or protocol family in use
- *Time to Live (TTL)* — Specifying the amount of time a record can be stored in cache before discarded.
- *Record Data* — Providing the type and class dependent data to describe the resources. This is a total different animal from the Record Type field since it actually displays the data value.

Exhibit 2-3 will provide short descriptions for some of the most common Resource Record types because you will need to understand them before jumping to do the magic with nslookup. If you have the time to spare, please feel free to download RFC 1034 and 1035 for complete information on all resource records.

Common Resource Records	Descriptions
A (Address)	Maps a host name to its corresponding IP address
CNAME (Canonical Name)	Defines an alias for the specified host name
HINFO (Host Information)	Provides software and hardware description of a host
MX (Mail Exchange)	Specifies mail exchange server(s) for a particular domain
NS (Name Server)	Provides a list of name servers for a domain
PTR (Pointer)	Maps an IP address to its corresponding host name
SOA (Start of Authority)	Designates the start of a zone and authoritative name server
TXT (Text)	Contains arbitrary text records associated with a host name

Exhibit 2-3. Common Resource Records

Congratulations, you have just gone through a really boring, yet indispensable part of nslookup. How would you expect to gather essential information about a remote host if you did not know how to properly direct `nslookup` to perform such tasks? Knowing the foundation and basic concepts of a DNS is what would make you feel more comfortable in constructing your `nslookup` queries, since you and the tool virtually speak the same language. Anyhow, it's time to turn on your computer and get to the command shell.

Since the command-line tool `nslookup` is readily available in most of Windows and UNIX platform; therefore, the upcoming sections will exclusively use `nslookup` to perform most of the queries. If `nslookup` did not come with your operating system for some reasons, you may use Sam Spade as a replacement. Sam Spade is a nifty network tool that includes a dozen of other useful network utilities

In order to get the most effective results from nslookup queries, it is advisable that you have the authoritative name server of the host in question. The authoritative name server of a host can be easily extracted from open source information or from the whois database.

There are two modes that you can use with `nslookup`, non-interactive mode and interactive mode. However, in practice, using non-interactive mode is not recommended since you are given limited options. Interactive mode of `nslookup` should be used if you want to experience the true power of `nslookup`.

In non-interactive mode, all you need to do is to append host name or IP address as an argument after the `nslookup` command and press enter. `Nslookup` will perform all the queries and then immediately terminate the session after displaying the returned results.

```
[bash]$ nslookup gotrice.com

Server:  default.name-servers.com
Address: 192.168.0.2#53

Non-authoritative answer:
Name:    gotrice.com
Address: 192.168.7.22
```

According to the example above, `nslookup` was started in non-interactive mode where the argument supplied was the host name of the computer — `you.gotrice.com` — and `nslookup` was smart enough to know that it was requested to perform a *forward name resolution*, or in other words, find the associated A record or IP address of this host name. The first two lines displayed by `nslookup` are the address of the name server used to perform the query. The first line is the fully qualified domain name of the server where in the second line is its corresponding IP address. The last two lines display the result of the query; in this case, reveal the IP address of the specified host name. Note how `nslookup` displayed "non-authoritative answer", that happens because the current default name server used to perform DNS query was not the authoritative name server of the corres-ponding domain `you.gotrice.com`.

Starting `nslookup` in interactive mode is trivial by executing `nslookup` command alone, without supplying it any arguments. Once you're in the interactive mode, you will be given a prompt and chances to tweak `nslookup` and enter any commands to your liking. `Nslookup` will evaluate all your queries, display the result, return you to the prompt, and await further commands or actions. You can exit `nslookup` if you explicitly type `exit` or `Ctrl-D` (`^D`) and you should remember that well because `nslookup` will not automatically do that for you. In all, `nslookup` has a lot of switches or options which you can use to manipulate `nslookup` queries once in the interactive mode, however the following list selectively provides you some of the most common and productive commands or options that you will frequently use:

- **`server`** — Designates `nslookup` to use the specified DNS server to query instead of the default name servers listed in /etc/resolv.conf
- **`host`** — Looks up for **A** and **PTR** record of a host
- **`set all`** — Shows current setting values, including current default name server and host
- **`set type`** — Changes to the specified resource record type. Refer to exhibit 2-3 for some common resource record types that you can use to make your `nslookup` queries more effective.
- **`set domain`** — Changes to the default domain name to the specified name.
- **`ls -t`** — Lists records of the specified record type. Exhibit 2-3 provides a listing of some common ones.
- **`ls -d`** — Lists all records for the domain. This is equivalent to the `ls -t ANY` command
- **`ls -a`** — Lists aliases of host in the domain. This is equivalent to the `ls -t CNAME` command
- **`ls -h`** — Provides hardware and software information for the domain. This is equivalent to the `ls -t HINFO`
- **`help`** or **`?`** — Self-explanatory.

Recent version of `nslookup` no longer supports the help and a few other commands. It is recommended that you use `dig` instead — which will be addressed shortly in the Bonus Material section.

The following shows some common usages of `nslookup` in interactive mode, such as querying the authoritative name server of a domain name for a certain type of record or performing a zone transfer:

```
[bash]$ nslookup
> server shady.shadow.com
Default Server:  shady.shadow.com
Address:  192.168.0.96#53
> set type=MX
> gotrice.com
Server:   shady.shadow.com
Address:  192.168.0.96

mail.gotrice.com preference = 10, mail exchanger = mail.gotrice.com
drdre.gotrice.com preference = 20, mail exchanger = drdre.gotrice.com
```

As already discussed previously, the only server that has a complete set of resource records or information related to a domain is the authoritative name server of that domain. Queries submitted to non-authoritative name servers of a particular domain will only yield limited and probably somewhat inaccurate information; thus, you need to first specify `nslookup` to use the authoritative one in order to get the most out of your queries.

The `server` option was used to specify `nslookup` use `shady.shadow.com` as the default name server for all forthcoming queries. The resource record type MX was set to acquire the server reveals the mail server of the target domain — `gotrice.com` — which as you can see, is `mail.gotrice.com`. This is just a simple example showing you how to use `nslookup` in interactive mode to perform a query for a specific type of DNS record; but perhaps the most interesting part that I want to show you is how easy it is to perform a zone transfer with `nslookup`.

```
[bash]$ nslookup
> server shady.shadow.com
Default Server:  shady.shadow.com
Address:  192.168.0.96#53
> set type=ANY
> ls –d gotrice.com

...<results truncated for brevity>

order                   1D IN A        192.168.0.192
drdre                   1D IN A        192.168.0.32
mail                    1D IN A        192.168.0.33
db2                     1D IN A        192.168.0.204
ttyl                    1D IN CNAME    nagging
dephunk                 1D IN A        192.168.0.69
                        1D IN HINFO    "1U-Rack" "Slackware"
                        1D IN MX 10    mail
                        1D IN NS       shady.shadow.com.
                        1D IN TXT      "Apache & MySQL"
dre                     1D IN A        192.168.0.21
                        1D IN HINFO    "admin" "win2k"
                        1D IN MX 20    drdre
                        1D IN TXT      "dre-owns-WarFTPd"
```

To instruct `nslookup` to perform a zone transfer, or in other words, collect all the records associated with a particular domain, you need to change the query record type to `any` along with the use of `ls -d` option. The HFINO and TXT records as you can see from above give away a lot of essential information for the two machines named `dre` and `dephunk` in the domain `gotrice.com`, including their respective IP address, operating systems, mail servers, and network daemons. This type of information is exactly what hackers and outsiders would normally love to lay their hands on before constructing a larger and more sophisticated attack against the target network.

It is worth noting that there is nothing wrong with a zone transfer itself because it provides a mean for secondary name servers to download and make a back up of all records stored in the primary name server to improve redundancy and load balancing. Had the primary name servers gone down or experienced any availability problems, the secondary name servers that have a copy of the zone in question will take place of the primary name servers to provide answers to those name resolution queries. The real problem of a zone transfer lies in which any name server with a sloppy configuration will happily provide a copy of the zone to anyone, not necessarily limited to authorized secondary name servers. Moreover, an unauthorized zone transfer from a name server that single-handedly stores and handles DNS records for both internal and external network will be far more devastating, since anyone who has a copy of the zone will also know the internal IP addressing information of the target network. Fortunately, in practice, unauthorized zone transfers are getting less common, but if you’re lucky, you might be able to find one of those in some poorly configured and least secured DNS systems.

COUNTERMEASURE: MINIMIZE EXPOSURE OF PUBLIC DNS RECORDS

Preventing leakage of sensitive information stored in the DNS server is not merely a sole responsibility of the clients, but the DNS administrators must also be accounted for since they are the one who assume total control over the server. A tiny bit of error in the name server's configuration can be just as fatal as when the clients supplying sensitive information into the DNS database.

To tighten up the security of the DNS server, it is incumbent upon the administrators to implicitly restrict zone transfer so that it would not be available to all machines but only to the specific and authorized ones. While normal DNS queries rely on UDP port 53 to communicate and pull the information from the server; zone transfers and oddly large DNS queries require the communication to be done over a more reliable transmission protocol — TCP — for transferring a larger amount of data, exceeding 512 bytes. Thus, placing the DNS server behind a firewall and instructing the firewall to deny all and allow only certain authorized IP addresses to access the DNS server through port 53/TCP is a good way to prevent all unauthorized zone transfer requests, but in so doing, you might inadvertently block legitimate DNS queries too. Nevertheless, it is important to configure the firewall to allow all inbound connections to port 53/UDP so that legitimate DNS queries can get through.

Newer versions of BIND (Berkeley Internet Name Domain), 8.2 and up, introduce a new security featured named Transaction Signatures (TSIGs), which is a cryptographic mechanism implemented to authenticate and secure DNS requests and responses. Moreover, keeping BIND or any other third-party DNS implementations up-to-date with the latest security patches helps prevent buffer overflows and other common security vulnerabilities that would allow the attackers to gain control of the servers upon exploitation.

It is important to apply principle of *least privilege* and separate DNS records for the internal and external network into two different DNS systems. Only then, even when an unauthorized zone transfer was successful, the impact and severity given by such would be greatly reduced, since the zone no longer disclosed internal IP address of the network to the attackers or anyone who had a hold of the zone files.

Restricting zone transfer is a great way to avoid providing hackers with unnecessary information that might simplify their wicked hacking tasks. However, putting the security and faith of your network onto the DNS administrator's lap is certainly not a good idea because security must be started from your own initiative too. As you can see from previous discussion of the DNS, there is certainly a lot of information, or so-called resource records, stored in the DNS database. It is up to you and your security policy to decide what type of resource record is implicitly necessary for the network to operate flawlessly and reliably; and in so doing, provide the DNS server with the necessary information accordingly. Avoid using unnecessary and informational record types such as TXT and HINFO will no doubt add more difficulties and obstacles to the hackers' journey. Information about internal IP addresses of the network must also be well concealed to prevent the attackers from learning and constructing the topology of the network. Even though limiting the amount of information goes into the DNS database cannot really help you prevent determined hackers from hacking the network; yet it is still better and worth the efforts to deter them by not making your network look like an attractive and easy target.

Finally yet importantly, if you take your DNS security seriously, you should regularly check the security of the authoritative DNS server by performing a zone transfer or interrogating the DNS server from the attackers' point of view, that way you will be able to decide exactly what should be revealed to the public and what should be not. Your organization must provide proper security training to the administrator who is responsible for configuring and administering the DNS server so that security oversight like zone transfer to unauthorized machines will not be missed. If controlling and configuring DNS server is not within your authority; then, choosing a reliable and well-secured DNS server at the first place does not seem to be a bad start either.

NETWORK ADDRESS RANGE

The last piece of information that hackers need to obtain in this footprinting phase is the address range of the network. Knowing the network address range of the target organization can be very helpful in which it helps the hackers calculate all possible entry points to the network, and more importantly, ascertains the hackers that they are indeed tackling the right target.

Finding all possible entry points to the network is important since even the most elite hackers will for sure experience the times when they are not able to hack into the desired network through a specific route, host, or machine. The only fix to that sort of situation is to have an option. Not being able to penetrate into the network by hacking the main host does not suggest that determined hackers will go after another target, because they can always compromise any accessible and less secured hosts of the target network to make their way in.

In real life, it is very often to see that heavy security countermeasures are deployed for crucial machines of the network or any machine that has the most exposure to the Internet, for instance, the web server that serve the main homepage. Ironically, security is not only about a security or safety state of a single specific host on the network, but it is the security of every other host on the network that really makes up a close-to-perfect security solution. For instance, the IIS (Internet Information Services) web server of the target might have various reasonably strong security mechanisms in place, such as eEye Secure IIS, Microsoft ISA, ISS BlackIce, and NGSEC Stack Defender to thwart most of the common attacks against the OS and web server, but it only takes one machine running vulnerable service on the network, such as un-patched RPCDCOM on Windows 2000, to subvert the security posture of the whole network.

Knowing the address range of the network not only helps the hackers find all possible accessible hosts of and entry points to the network, but also helps the hackers avoid hacking the wrong target. Hacking the wrong target may lead the hackers to undesirable results because perception about security is not the same for everyone. Some do take security seriously, while some just don't really care, and for this reason, if the hackers are mistakenly hacking into a highly security-cautious organization, chances for them to be detected and caught right on the spot are clearly inevitable. Imagine what would happen when the hackers spend countless efforts and times to hack fbi.gov instead of i-am-really-your-fish-dude.com? Of course, hacking an "unattractive" target like that certainly is not a good thing. Come on, messing with the FBI? Anyone?

Now that you know how important it is to locate the right address range and associated subnets of one network, it is now time to see where you can find such information.

There are typically two methods for locating the address range of the network, one is to consult with the four RIRs since they are responsible for providing and managing IP address resources; thus, it is obvious that their databases must store that kind of information. The second method, *trace route*, in its truest sense, is not a method of which you can use to locate network range of the target. Trace route is a method where hackers attempt to send various network packets from their computers to the destination network to determine the path to the network and probe for all possible routers or firewalls belong to the network. Based on the outcomes of trace route, the hackers may be able to deduce the address range of the network; however, that requires quite a bit of works.

The following section will use the American Registry for Internet Numbers (ARIN) exclusively to discuss how one can locate network address range for the .com, .net, and .edu domain names.

:.::.: ARIN (AMERICAN REGISTRY FOR INTERNET NUMBERS)

ARIN is responsible for allocating, distributing, and managing Internet numbering resources of the Northern America and sub-Saharan Africa region. Information related to registered resources under ARIN, including IP address space v4 and v6 or autonomous system numbers, can be easily extracted from the ARIN's whois database by using `whois` from the UNIX command shell or the web-based whois client provided by ARIN.

From previous discussions, you have been shown how one can submit queries to the ARIN whois server to obtain registration details for a particular network block or to mine for any subnets associated with a network. Thus, this section will not reintroduce everything all over again but rather provide you some additional useful resources and tips. If you have already forgotten how you can identify associated network blocks or address range of the target network with ARIN's whois queries, please consider revising various whois query types that have been discussed in the whois section — page 17.

Exhibit 2-4 lists all whois flags that are accepted by ARIN's whois server. You can use any of these flags combining with your whois queries to make the queries flexible and yield more productive results. Note, you can easily retrieve this sort of information anytime by submitting `help` or `?` to the ARIN's whois server.

Exhibit 2-4. Accepted Whois Flags (Excerpted from ARIN's whois response)

Query-by-record-type:

To limit your query to a specific record type, include one of the following flags:

n	Network address space
a	Autonomous systems
p	Points of contact
o	Organizations
c	End-user customers

Query-by-attribute:

To limit your query to a specific record attribute, include one of the following flags:

@ <domain name> Searches for matches by the domain-portion of an e-mail address
! <handle> Searches for matches by handle or ID
. <name> Searches for matches by name

Searches that retrieve a single record will display the full record. Searches that retrieve more than one record will be displayed in list output.

Display flags:

To modify the way that the query results display, include one of the following flags:

+ Shows detailed (aka 'full' output) display for EACH match
- Shows summary only (aka 'list' output), even if single match returned

The + flag cannot be used with the sub-query feature described below.

Record hierarchy:

Records in the ARIN WHOIS database have hierarchical relationships with other records. To display those related records, use the following flags:

< Displays the record related up the hierarchy. For a network, displays the supernet, or parent network, in detailed (full) format.
> Displays the record(s) related down the hierarchy. For a network, displays the subdelegation(s), or subnets, below the network, in summary (list) format. For an organization or customer, displays the resource(s) registered to that organization or customer in summary (list) format.

Wildcard queries:

WHOIS supports wildcard queries. This feature is only supported as a trailing character option. To take advantage of this append the query with an asterisk (*). This can also be used in combination with any flags defined above.

As already stated, ARIN provides a web-based whois client which you can freely use to gather and extract records stored in the whois database. The whois client is available at http://www.arin.net/whois/index.html. In addition, ARIN provides free online Computer Based Training showing ARIN's whois users how to properly submit a whois query and interpret the outputs. The course is intended for both novice and experienced whois users and it is available at http://www.arin.net/library/training/WHOIS_CBT/index.html.

:.::.: TRACEROUTE

Similar to `nslookup`, `traceroute` is an indispensable debugging tool used by network engineers and administrators world wide to troubleshoot the network in the event of problems. The tool comes by default with almost every UNIX and Linux distributions under the name `traceroute`, and for Windows platform, the tool is named just a slightly bit different, `tracert`.

Unless the two computers are directly connected via a null modem cable or Ethernet interface, it is with certainty that any packet exchanged between the two will be routed through several places. `Traceroute` is, in short, a smart network utility that can be used to determine the path or locations that the packet had to go through before reaching its final destination.

Right after a computer sent off a packet, the computer will no longer have any control or responsibility over the packet and the best it can do is only to "sit and pray" for the packet arriving at the right place. It is, indeed, the ultimate responsibility of all routers around the world to make a wise decision and deliver the packet to its intended recipient. In an optimistic situation where the target network is alive and kicking, the packet will be happily delivered at once with no real concerns. However, from a pragmatic standpoint, the Internet structure is just too huge and complicated that errors are inevitable; and for this reason, the packet might not always be able to reach the target network. As a result, if the routers did not discard and kept circulating the defective packet around until it reaches its intended destination; the Internet would just well collapse on its own weights within hours. Fortunately, the "Internet gods" did not want such miserable situation happen to the much-loved Internet; therefore, they came up with a solution called *Time to Live*, which is implemented into the header fields of the Internet Protocol (IP) as you see these days.

When a packet is sent from one computer to another, the *Time to Live* (TTL) field in the packet header is set to a default value, defining a maximum number of *hops* a packet is allowed to have to prevent the packet from looping endlessly in the event of routing problems. A hop literally means when a packet leaps from one router to another. Each router that handles the packet will have to decrease the defined maximum hops, or TTL value by one, before forwarding the packet to the successive router that might know something about the packet destination. The packet will be kept forwarding until it reaches the target network or until the value in its TTL field is zero; whichever comes first. If the value in the TTL field is zero, the router will stop routing the packet and send an *ICMP* (*Internet Control Message Protocol*) error message to inform the originator.

"Fine, now I know what TTL is but what does this have to do with `traceroute`?" `Traceroute` exploits the convenient feature of the TTL field to determine the path a packet must take to reach the target network. First, `traceroute` will purposely send a packet with a TTL value of one. The first router receiving this packet will decrease the TTL value by one and because the TTL value of the packet after the decrement is 0; therefore, the router will send out an error message to inform the sender regarding this "unthinkable error". Upon receiving the error message, `traceroute` records the IP address of the first router, and if possible, translates the IP address of the router to its fully qualified domain name. With the exact same technique, `traceroute` sends subsequent packets with the TTL value of two, and then three, and so forth until a complete network path between the originating computer and destination computer is revealed, or until the maximum number of hops set by traceroute is reached — normally 30 hops.

Enough with all the theory and “behind the scenes”, the following example gives you some ideas how traceroute works in real life. Note for Windows users, traceroute in UNIX platform and tracert provide the same concept and functionality, please use the tool tracert accordingly.

```
[bash]$ traceroute gothere.com
traceroute to gothere.com (192.168.7.22), 30 hops max, 38 byte packets

1  out (172.16.2.21) 0.973 ms  0.930 ms  0.864 ms
2  172.18.23.181 (172.18.23.181)  5.781 ms  5.757 ms  5.668 ms
3  go.beyond.net (172.31.32.33)  24.443 ms  6.136 ms  5.991 ms
4  bash.hoondie.com (192.168.99.32)  47.821 ms  47.502 ms  47.569 ms
5  enr1.townhall.org(192.168.43.3)  47.461 ms  47.334 ms  47.446 ms
6  exrtr.layer42.net (192.168.19.22)  55.951 ms  47.476 ms  47.451 ms
7  192.168.10.21 (192.168.10.21)  65.297 ms  60.195 ms  47.669 ms
8  finally.gothere.com (192.168.7.22)  47.735 ms  47.384 ms  60.536 ms
```

When you analyze traceroute output, you should perceive the information from the bottom to the top. Normally any information provided in the first part of output of traceroute is gibberish, your goal is to gather essential information related to the target remember? So it is imperative and wise that you look for hops provided in the very last part of the output since only those last hops with higher TTL values can actually reach the target. The first few hops with low TTL values of two or three only identify intermediary devices or routers on the path, which may be, or may be not, important to some people. The last hop from the traceroute output should reveal information about the specific computer that you wanted to know more in the first place.

In a commonly seen setup of a basic network, there should be an external router to allow incoming traffics to the internal network, and an internal router, or a firewall, or packet-filtering device to validate and route the traffics to the right machine in the network. As a result, if the last hop shown in your traceroute output is the destination machine, then the second and the third to the last hop should be the internal and external router of the network respectively.

However, that judgment is based on a very optimistic situation and it probably will not work toward a complex and large-scale network environment because the network may have multiple connections to the Internet, as well as multiple paths to the destination machine. Therefore, in order to get a complete picture of the network, such as its entry points and the location of routing devices, it is imperative that you must traceroute to various different systems located on the network.

In the above traceroute output, everything went smooth as no packet was dropped on the way. The last hop is the destination machine, finally.gothere.com with the IP address of 192.168.7.22. The 6th and 7th hop should be the external and internet router of the network respectively. Remember, you can never be so sure about that and therefore you have to perform traceroute a few more times to other systems of the network in order to ascertain your assumption.

Given that any machine located behind the same external router is on the same network, you should perform `traceroute` to a few other machines to see whether the packets destined for those machines actually go through the same external router. Continue with the above example, if you know that the IP address of finally.gotrice.com is 192.168.7.22 and its external router is 192.168.19.22, then you should perform two different `traceroute`, one is to any system of the 192.168.6.0 network and the other is to any system of the 192.168.8.0 network. If the packets destined for those systems are not going through the same external router as the 192.168.7.22, then it is safe to say that finally.gotrice.com has nothing to do with the 192.168.6.0 and 192.168.8.0 network. However, knowing only that is not good enough because you have yet known the real address range of finally.gotrice.com; therefore, you need to `traceroute` to a few more systems located on the 192.168.7.0 network, or particularly, `traceroute` to the first and the last available systems of 192.168.7.0 and use the same method to analyze and deduce the network address range. Practically, using `traceroute` to find the address range of the network is not recommended since it involves in many additional different steps and the information revealed may not always be accurate. It is recommended that you consult with the four RIRs for such information.

COUNTERMEASURE: ICMP & NAMING CONVENTION

Strictly speaking, there is no single definitive method that you can use to effectively block `traceroute`. The utility and its underlying mechanism rely heavily on a crucial messaging protocol — ICMP — in order to work and function properly; and under that impression, many people seems to think blocking ICMP is the key to solve the problem. However, there are indeed many useful ICMP-based network utilities created to help the administrators troubleshoot the network in the event of problems; and consequently, blocking ICMP might not be the best solution since it may entail many other side-effects. In any case, if your security policy permits blocking ICMP at the external router and there is no real impact upon the network from such act, then blocking ICMP will not be a bad idea at all, because it will greatly reduce a remarkable amount of ICMP-based attacks or network probes against the network. Note, `traceroute` and many other `traceroute`-like tools are capable of sending UDP packets in lieu of the traditional ICMP packets; hence, blocking ICMP is certainly not as effective as it seems. Besides, address range of the network can always be revealed by submitting query to the ARIN's whois database.

Another solution that may help prevent hackers from constructing your network topology is to avoid using informational naming convention for your access control or routing devices. Although a descriptive name describing the location and function of a device can be useful for troubleshooting tasks, hackers also find such naming convention useful in helping them locate and identify the role of each device shown in the `traceroute` path.

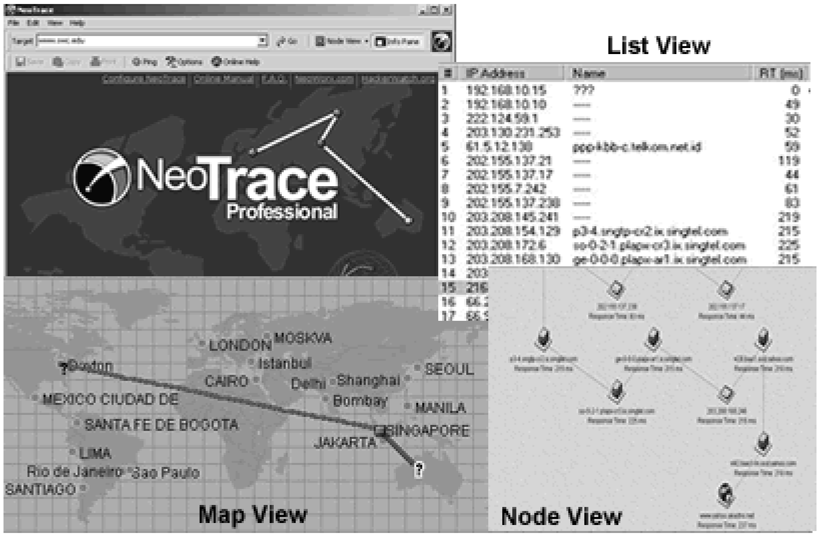
Now that you know how to use `whois`, `nslookup`, and `traceroute` to perform your queries as well as to map the target network, it's time to look at some of the different tools that can help you automate and speed up the process through the Graphical User Interface (GUI).

NEOTRACE

For those who hate carrying out `traceroute` from the ugly black console, who are too lazy to perform additional steps of whois queries to verify the network owner and geographical location of each node that `traceroute` packets went through, NeoTrace is the perfect solution. Welcome to the pretty face of `traceroute`.

NeoTrace is a diagnostic network tool widely used to serve for the same purposes as `traceroute` and perhaps even more in some extent. While `traceroute` is only capable of showing you the IP address, and if applicable, domain name, of each node along the path, NeoTrace takes that to another level by automatically tracing and showing you all information about the owner, registration details, network block, fully qualified domain name, and geographical location of each node through an intuitive interface. Thus, it is easy to understand why many people often look at NeoTrace as a combination of `whois`, DNS resolvers, and `traceroute` in one visual package.

Figure 2-3 shows a screen shot of NeoTrace output with three different views, node view, map view, and list view. Map view provides a nice graphical output of `traceroute` in which users can easily identify the geographical location of each node through the detailed world map. Node view provides you with an abstract view of the trace path and shows the logical relationships between each node. List view offers you the same kind of information that you normally see from `traceroute` outputs including node address, domain name, and time response.



#	IP Address	Name	RT (ms)
1	192.168.10.15	???	0
2	192.168.10.10	——	49
3	222.124.59.1	——	30
4	203.130.231.253	——	52
5	61.5.12.130	ppp-kbb-c.telkom.net.id	59
6	202.155.137.21	——	119
7	202.155.137.17	——	44
8	202.155.7.242	——	61
9	202.155.137.238	——	83
10	203.208.145.241	——	219
11	203.208.154.129	p3-4.sngtp-cr2.ix.singtel.com	215
12	203.208.172.6	so-0-2-1.plapx-cr3.ix.singtel.com	225
13	203.208.168.130	ge-0-0-0.plapx-ar1.ix.singtel.com	215
14	203		
15	216		
16	66.2		
17	66.3		

Figure 2-3. NeoTrace with 3 different views

Equipped with all the enhanced features and providing a great amount of information, NeoTrace undeniably plays an important role in helping the law enforcement as well as Internet professional trace down the location of the Internet abusers or source of spam emails. Yet there is no such free meal in this world as the tool comes with a reasonable price of $29.95. NeoTrace is only available for Windows users and you can download a 30-days trial of NeoTrace at http://www.neoworx.com. Remember, you can get the same detailed level of information provided by NeoTrace with freely available network tools like `traceroute`, `whois`, and `nslookup` or `dig`.

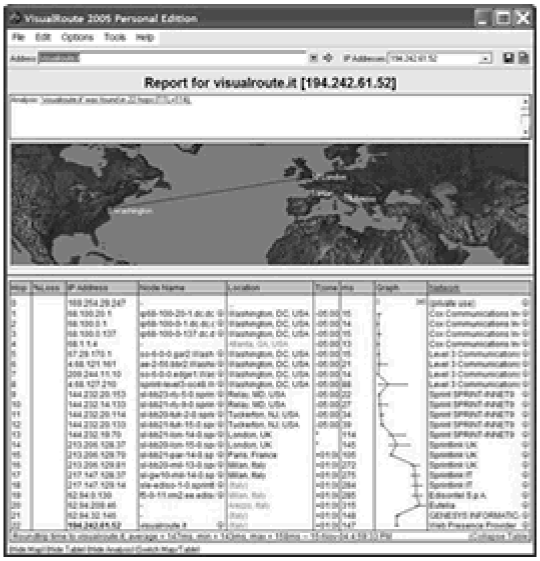
Instead of spending 10 mind-numbling pages showing you how to use Neo Trace or any other hacking tool mentioned in the C|EH course outline, the book only provides you a brief introduction of what the tool does, its usage and functionality. For the C|EH exam, you will only need to know the purposes of the tools and under which circumstances they may be used for. Nevertheless, a thorough explanation will be provided only when the tool is exceptionally popular and necessary to complete the task, like **whois** or **nmap**.

VISUALROUTE TRACE & EMAIL TRACKER

VisualRoute is another similar diagnostic and investigative network utility like NeoTrace. The tool incorporates all the strong features from many freely available command-line network tools like `traceroute`, `whois`, and `nslookup` to provide users a great deal of information in term of quantity and quality through the graphical interface. However, unlike NeoTrace, VisualRoute has two available editions, a Personal Edition and a Server Edition that provides VisualRoute services to remote users. Both of the editions are compatible with many popular platforms, including Linux, UNIX, and Windows. Note that VisualRoute is not merely a GUI version of `traceroute`; in fact, it does provide some interesting features, rather than just trace and display. Figure 2-4 shows a screenshot of VisualRoute Trace in action. The captured screenshot was taken from the official website of VisualRoute, http://www.visualroute.com.

VisualRoute provides a feature called “Scan Network” listed under the Options menu, helping you determine whether the host in question was actually unreachable or whether its entire network was unreachable in the event VisualRoute can’t receive any response from the monitored host. This feature is actually derived from the infamous `ping` utility that comes with almost, if not all, OS platforms out there. When the feature Scan Network is selected, VisualRoute will aggressively ping the entire class C network, which the monitored host belongs to, until receiving a response from any host or until the scanning is done. As you can clearly see, this scan feature attempts to make a direct contact between the target network and the tool’s users, and thus, is relatively aggressive. For this reason, great care must be given before considering using this sort of technique as you may tip off intrusion detection systems and firewalls at the target network about your presence and reconnaissance attempts.

In addition, VisualRoute also reveals the software and version of the web server. However, the tool does not always yield accurate information since the technique used to detect information about the web server relies solely on the banner included in the server's response, which can be easily manipulated by clever administrator to disguise the real software and its version. This discovery technique is also normally referred as "banner grabbing". Yet again, you can do this manually with many free tools out there, such as `telnet`, `netcat`, or Netcraft.


VisualRoute 2005 Personal Edition
File Edit Options Tools Help
Address
IP Address

Report for visualroute.it [194.242.61.52]

Analysis: "visualroute.it" was found in 22 hops (TTL=114).

London
Washington
Paris
Milan

Hop	%Loss	IP Address	Node Name	Location	Tzone	ms	Graph	Network
0		169.254.29.247	-	-				(private use)
1		68.100.20.1	ip68-100-20-1.dc.dc	Washington, DC, USA	-05:00	15		Cox Communications In
2		68.100.0.1	ip68-100-0-1.dc.dc.c	Washington, DC, USA	-05:00	14		Cox Communications In
3		68.100.0.137	ip68-100-0-137.dc.d	Washington, DC, USA	-05:00	15		Cox Communications In
4		68.1.1.4	-	Atlanta, GA, USA	-05:00	13		Cox Communications In
5		67.29.170.1	so-6-0-0.gar2.Wash	Washington, DC, USA	-05:00	15		Level 3 Communications
6		4.68.121.161	ae-2-55.bbr2.Washi	Washington, DC, USA	-05:00	21		Level 3 Communications
7		209.244.11.10	so-6-0-0.edge1.Wa	Washington, DC, USA	-05:00	14		Level 3 Communications
8		4.68.127.210	sprint-level3-oc48.W	Washington, DC, USA	-05:00	88		Level 3 Communications
9		144.232.20.153	sl-bb23-rly-5-0.sprin	Relay, MD, USA	-05:00	22		Sprint SPRINT-INNET9
10		144.232.14.133	sl-bb21-rly-9-0.sprin	Relay, MD, USA	-05:00	27		Sprint SPRINT-INNET9
11		144.232.20.114	sl-bb20-tuk-2-0.sprin	Tuckerton, NJ, USA	-05:00	34		Sprint SPRINT-INNET9
12		144.232.20.133	sl-bb21-tuk-15-0.spr	Tuckerton, NJ, USA	-05:00	39		Sprint SPRINT-INNET9
13		144.232.19.70	sl-bb21-lon-14-0.sp	London, UK	*	114		Sprint SPRINT-INNET9
14		213.206.128.37	sl-bb20-lon-15-0.sp	London, UK	*	145		Sprintlink UK
15		213.206.129.70	sl-bb21-par-14-0.sp	Paris, France	+01:00	105		Sprintlink UK
16		213.206.129.81	sl-bb20-mil-13-0.sp	Milan, Italy	+01:00	272		Sprintlink UK
17		217.147.128.37	sl-gw10-mil-14-0.sp	Milan, Italy	+01:00	275		Sprintlink IT
18		217.147.129.14	sle-ediso-1-0.sprintit	(Italy)	+01:00	284		Sprintlink IT
19		62.94.0.130	f5-0-11.mi2.ea.ediso	Milan, Italy	+01:00	285		Edisontel S.p.A.
20		62.94.208.46	-	Arezzo, Italy	+01:00	315		Eutelia
21		62.94.32.146	-	(Italy)	+01:00	148		GENESYS INFORMATIC
22		194.242.61.52	visualroute.it	(Italy)	+01:00	147		Web Presence Provider

Roundtrip time to visualroute.it, average = 147ms, min = 143ms, max = 158ms -- 15-Nov-04 4:59:33 PM

Figure 2-4. VisualRoute Trace

Another neat feature included in this multi-purposes network utility is eMailTracker. This neat little utility helps you identify the mail server of the domain name or email address in question. Identifying the correct mail server of a particular domain is essential because not all mail servers necessarily begin with *mail.* but it can also be *butterfly.* or even *rabbit.* to that matter. After the mail server had been identified, the tool then helps uncover information associated with that mail server, including its respective IP address, fully qualified domain name, software, and version information.

The information provided by eMailTracker, of course, relies heavily on the results of various DNS queries as well as banner grabbing technique. First, it submits a NS record type query to the root DNS server to collect information about the authoritative name servers of the domain. Using the information provided by the root DNS server, eMailTracker yet again submits another query to the authoritative name server to look for the MX (Mail eXchange) record of the domain. After knowing the address of the mail server, the tool then attempts to make contact to the mail server; and subsequently, analyze the information included in the server's response to look for clues.

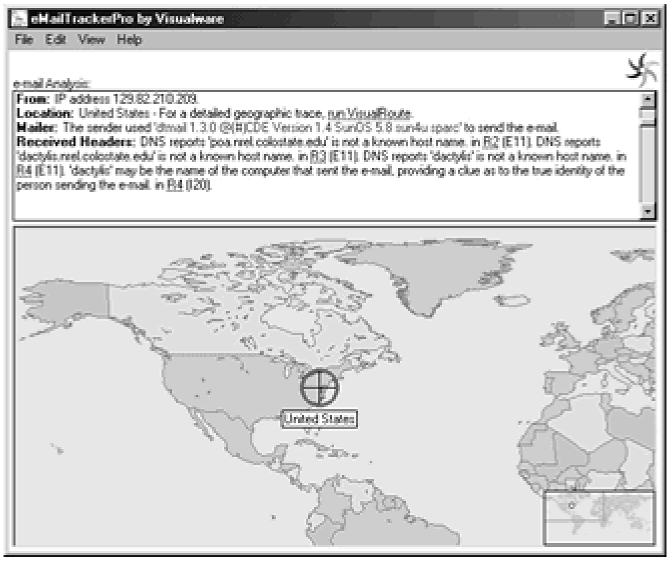
Unless you have specific needs to use VisualRoute, or NeoTrace, or any other graphical network utilities of the same genre, it is suggested that you use those free command-line tools as introduced earlier to perform footprinting. Using those command-line tools to manually gather information about the target organization will greatly improve your knowledge and footprinting skills, since those tools all require you to know what's going on under the hood before you can actually use them effectively to get the most desirable result. Just keep this in mind, you can use those free tools and obtain the same amount of information as provided by NeoTrace or VisualRoute while at the same time save $30 for your KFC. Anyhow, if you have the money to spare and wish to speed up the footprinting process, then purchasing VisualRoute or Neo-Trace is not a bad choice either, since those tools come with a reasonable price as well as many neat features.

You can download trial version of VisualRoute at http://www.visualroute.com.

EMAILTRACKER PRO

The tool eMailTracker Pro is the stand-alone version of the basic email-tracking tool included as part of the VisualRoute product. This stand-alone version is far more superior packed with many nifty tracking tricks.

First of which, not only does the tool help you uncover information about the mail server of an email address but also assist you in tracking down the sender of the email. By providing the tool with the full header of an email address, a lot of information about the origin of the email can be revealed, including the mail client, time and date the email was composed, the originated IP address, the location of the sender, and the involved mail servers. Figure 2-5 illustrates an instance of eMailTracker Pro.



From: IP address 129.82.210.209.
Location: United States - For a detailed geographic trace, run VisualRoute.
Mailer: The sender used 'dtmail 1.3.0 @(#)CDE Version 1.4 SunOS 5.8 sun4u sparc' to send the e-mail.
Received Headers: DNS reports 'poa.nrel.colostate.edu' is not a known host name. in R2 (E11). DNS reports 'dactylis.nrel.colostate.edu' is not a known host name. in R3 (E11). DNS reports 'dactylis' is not a known host name. in R4 (E11). 'dactylis' may be the name of the computer that sent the e-mail, providing a clue as to the true identity of the person sending the e-mail. in R4 (I20).

Figure 2-5. eMailTracker Pro

Because information provided in an email header can be forged and spammers always attempt to use forgery information to cover their true identities; therefore, the tool is also equipped with many different techniques that can be used to parse and validate the authenticity of the information provided in the header. But the one validating technique that is worth to bring to your attention is the heavily usage of DNS lookups.

When an email header is pasted for parsing, eMailTracker Pro will attempt to track down the original location by looking for and analyzing the most important field in the header, the *Received* field. The Received field provides a wealth of information about the locations or sites that an email went through before reaching its intended recipient. However, such information the Received field can also be forged. The spammers can always inject bogus information, such as fake domain name or host name of the machine, in an attempt to misdirect cautious users; and it is the responsibility of eMailTracker Pro to reveal those forgery attempts and get the right information. Firstly, the tool will attempt to perform a *reverse DNS lookup* based on the IP address found in the Received field. A reverse DNS lookup is an attempt to map an IP address to its corresponding host name, and if you can recall correctly of what have already been discussed, it is a DNS query of the PTR (Pointer) record type. After the host name of the IP address is revealed, the tool then performs an A (Address) record query type to look for the associated IP address of that host name. Finally, the tool compares the two DNS query results and if either one of them is mismatched, the tool will notify you regarding that possible forgery attempt as well as offer you potential information about the true identity of the sender — which is of course based on the IP address stored in the Received field since it is the least likely to be forged.

An email header gives away a lot of information about the sender and it certainly has a lot more fields rather than just the Received field alone. In the Bonus Material section, you will be shown how to analyze an email header manually without the need of using any other tools.

You may already wonder if this email-tracking tool has anything to do with footprinting at all. Your doubt is right to some extent because eMailTracker Pro is popularly used by the law enforcement and professionals to track down the source of spam emails and it is certainly has nothing to do with footprinting there. However, from an attacker viewpoint, an attacker could also use this tool for a total different purpose. By sifting through those USENET posts and looking for any email originated from the target network, the attacker can easily extract the email header and paste it to eMailTracker for analysis. Upon completion of the analysis, it should be trivial for the attacker to reveal information about the mail server, internal IP addressing scheme, as well as possible architecture of the target network; and thereby, levering the attack.

VISUAL LOOKOUT

VisualLookout is probably the oddest product mentioned in this footprinting phase because it absolutely has nothing to do with the hacking or information gathering, but rather monitoring network connection and detecting intrusion attempts. It will be briefly discussed here just for the sake of completeness.

VisualLookout is another fine product from VisualWare, which provides a measure to help one monitor connection activity on up to 100 systems in real time. The AutoSentry feature included in the tool allows users to specify and exclusively monitor specific ports, domain names, or IP addresses. Users are also given option to set a threshold for a certain type of network activity that need to be monitored and when the defined threshold is reached or crossed, the tool will appropriately make contact to the users by sending an email, running an application, displaying a popup, or sending SNMP trap.

More information about VisualLookout is available at http://www.visuallookout.com.

SMARTWHOIS

Originally built to address the shortcomings of standard whois utilities, SmartWhois provides many additional enhanced features that make your whois queries as pleasant and efficient as possible.

The current version of SmartWhois, 4.0, provides support for a wide range of different domain names including country-code top-level domains as well as many other new and less popular international domains, such as .uk, .fr, .info, .biz, .aero, .coop and so forth. On top of that, the application also adds plug-in to Microsoft Outlook and Internet Explorer upon installation, allowing one to perform whois query of IP addresses or domain names directly from those applications.

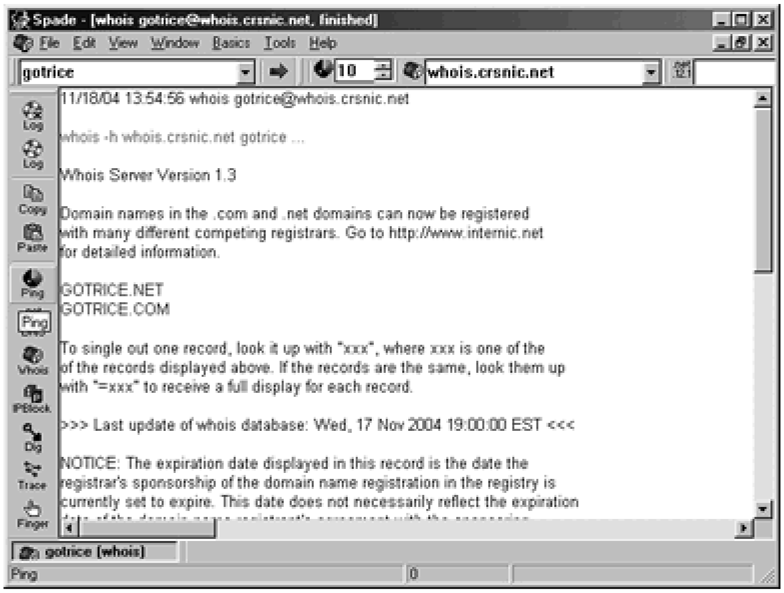
Simply inputting the domain name that you need to lookup into the search box, SmartWhois will automatically select the most appropriate server and attempt to retrieve information from more than 60 different whois servers all around the world. Performing whois query with SmartWhois will definitely save you a considerable amount of time since you only need to submit one type of whois query and SmartWhois will do everything to bring you the right information, which you initially asked for.

Another advantage of using SmartWhois is probably the allowance of saving the obtained information to an archive file. Anytime you start SmartWhois and load the saved archive file, you will be given a chance to update the file should the records associated with the domain names or IP addresses listed in the archive file have been changed since your last queries. On top of that, the tool also allows saving the results into several different formats, such as, XLS, HTML, TXT and XML.

SmartWhois also has a few command line options to facilitate the batch processing. Batch processing lets you supply SmartWhois with a text or batch file that contains IP addresses or domain names that you need to query on and have SmartWhois successively query all the domains and IP addresses listed in the batch file, one by one.

SAM SPADE

According to the introduction of Sam Spade in the help file, Sam Spade is “a general-purpose Internet utility package, with some extra features to help in tracing the source of spam and other forms of Internet harassment.” However, as I have said about hacking tools, they are like double-edged swords, which can be used to harm or to protect you. In the real world, hackers are of course not going to use Sam Spade to trace spam or other forms of Internet harassment, but almost certainly, they will use Sam Spade to assist them in gathering more information about the target network, and subsequently, using the collected information to further harass and hack the target. It's sad, but true fact. Figure 2-2 illustrates a whois query performed through Sam Spade.



```
11/18/04 13:54:56 whois gotrice@whois.crsnic.net

whois -h whois.crsnic.net gotrice ...

Whois Server Version 1.3

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

GOTRICE.NET
GOTRICE.COM

To single out one record, look it up with "xxx", where xxx is one of the
of the records displayed above. If the records are the same, look them up
with "=xxx" to receive a full display for each record.

>>> Last update of whois database: Wed, 17 Nov 2004 19:00:00 EST <<<

NOTICE: The expiration date displayed in this record is the date the
registrar's sponsorship of the domain name registration in the registry is
currently set to expire. This date does not necessarily reflect the expiration
```

Figure 2-6. Perform Whois Query with Sam Spade

In general, Sam Spade is a great single network tool featuring many other neat network utilities, including whois, DNS lookup, dig, ping, traceroute, and finger. You can easily access to those basic network tools by selecting any one of the icons listed on the toolbar on the left hand side. Additional advanced network tools, such as zone transfer, SMTP relay check, web site crawling, and URL decoding are also available under the Tools menu; however, those tools all require a little tweak before you can actually start using them.

All of the network utilities included in Sam Spade can be interacted by the users through a friendly interface, the intuitive GUI (Graphical User Interface), allowing easy seeking, discovering, and analyzing information about a network. Using the tool itself is also an easy part, spending ten minutes playing around with it and you will sure be fine.

SUMMARY

Footprinting is the very first phase of hacking which covers two broad categories of information that hackers can easily obtain: open source information and network address range. Open source information is any information that can be accessed publicly by anyone without any restriction. In this footprinting phase, the hackers will only need to use non-intrusive information gathering techniques to profile and collect information related to the target organization's network; thus, the target would not be aware about such reconnaissance attempts. Some of the tools and techniques that can help the hackers gather open source information include whois queries, DNS queries, search engines, job offer advertisement, or even the target's website. After gaining some basic knowledge of the target from those obtained open source information, the hackers consult with the RIRs (Regional Internet Registries) or use `traceroute` to subsequently identify the address range of the target network. Finally, the chapter ends with introduction of several commercial network tools that incorporate many functions similar to those found in `nslookup`, `whois`, `and traceroute` into a visual package. In sum, hackers who have taken necessary steps to accomplish this footprinting phase will have a unique profile of the target network, which in turn, can be used as a base for the hackers to advance to the next phase or category of information gathering — Scanning.

BONUS MATERIAL

As already stated, this Bonus section will provide you some alternative security tools and techniques that can be used to accomplish some of the tasks mentioned in this chapter. They may not be the latest but I have found them to be useful and sometimes yield more productive results than those tools mentioned earlier in the chapter. You will not need to read this to prepare for the C|EH exam. This is only useful for those who want to learn and think "outside the book".

DIG (DOMAIN INFORMATION GROPER)

`Dig` is a great network utility that hackers should not underestimate its capability while they are still in this footprinting stage. The sole purpose of using `dig` is to interrogate the DNS server. You heard it right, interrogate as in its truest sense, asking the DNS server to spit out more information about the target network, just like `nslookup`. However, `dig` is just not as prevalent as nslookup and if you are not be able to find it within your Linux or UNIX distribution then you can visit www.isc.org.

> Admittedly, configuring DNS servers properly and securely is a complicated process, and it is quite often to see the administrators decide to go for the functionality and usability instead of security, which of course, is a bad practice since it greatly simplifies the hackers' tasks. I once knew a guy through an IRC channel who went by the pseudonym "rabbitpulse" showing me how he was able to query the authoritative DNS server of the biggest ISP in Vietnam and retrieve a huge amount of information related to the associated hosts and domain names for that ISP, simply by using `dig`. It is more likely that he was able to carry out a zone transfer due to poor configuration.

`Nslookup` and `dig`, in an essence, might be no difference since they are both used for the purposes of querying and debugging DNS related problems. However, the most obvious difference between the two DNS debugging tools is in the way each tool presenting the results to the users. While `nslookup` provides a trimmed version of a DNS message response for concision, `dig` by default shows a verbose output, which is a complete format of a DNS message including the *header, question, answer, authority,* and *additional* sections. Another notable difference lies in how dig is only capable of working in non-interactive mode, which means you have to supply all the arguments on the same command line as dig. Even so, dig is just as useful as nslookup. It has an intensive set of switches and options that offers you the opportunity to tune and tweak your DNS queries to whatever your liking for the best effects. For those who still love nslookup, nslookup has been around for too long and it is about time for it to concede the crown to the later rival — dig — which is more flexible and advanced.

You must enter the host name, the resource record type that you wish to query, and the authoritative name server as the arguments following dig on the same command line. Those arguments can be placed in arbitrary order and dig will still be able to get the job done. If you're just running dig without supplying it any argument, dig will just use all the default values to perform DNS query for the local host; that is using the name servers defined in /etc/resolv.conf and resource record type A (Address).

The following shows you some examples and common tasks that you would normally use dig to gather more information about the target network. Consider looking up dig's man page if you are having troubles using the tool or getting it works.

As usual, you have to know the authoritative name server of the domain name before you can make any further queries. Mind you that the domain gotrice.com is fictitious and it was created to be only accessible within a mock network environment setup for demonstration purposes; thus, all of its associated information as you will see below is also irrelevant to the real Internet domain gotrice.com. Do *not* use the query results provided in this section to attack or hammer the real network because they are entirely unrelated. You have been warned.

```
[bash]$ dig @a.root-servers.net gotrice.com ns +nostats

; <<>> DiG 9.2.2 <<>> @198.41.0.4 gotrice.com ns
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 27349
;; flags: qr aa rd; QUERY: 1, ANSWER: 0, AUTHORITY: 3, ADDITIONAL: 3

;; QUESTION SECTION:
;gotrice.com.                       IN    NS

;; AUTHORITY SECTION:
gotrice.com.            86400       IN    NS    shady.shadow.com.
gotrice.com.            86400       IN    NS    no.shadow.com.
gotrice.com.            86400       IN    NS    wutang.clan.org.

;; ADDITIONAL SECTION:
shady.shadow.com.       86400       IN    A     192.168.0.96
no.shadow.com.          86400       IN    A     192.168.0.97
wutang.clan.org.        86400       IN    A     192.168.0.98
```

The top part and the last part of a dig output usually provide additional information about the query such as query statistics, comments, total time taken, size of the reply, and so forth. However, the option +nostats is used to specify dig not to print the query statistics, which should have been tailed after the additional section, to save space.

The *Header* section located at the top part of the output offers essential information telling you what to expect in the DNS message responses. The Q*uestion* section typically recites what your query originally was. There is no *Answer* section in the above query because this query is just a referral query, which asks a name server to provide information about other name servers. All answers related to the referred name servers are listed in the *Authority* section. The *Additional* section provides extra information that dig think you might also want to know. Obtaining this kind of information doesn't hurt either, it is always good to have the bonus don't you think?

Practically, you don't need to perform many different types of queries just to get the desired answer because dig is also capable of transferring or directing your queries to the most appropriate server that might know something about what you are asking, similar to the redirection feature of the command-line tool whois. You just need append the option +recurse to any of your record query.

After the authoritative name server of the domain is known, you then can specify dig to use the authoritative name server to perform further query for the domain gotrice.com. For even more brevity, subsequent queries will include the +nocmd and +noquestion to disable printing initial comments and question section of the query altogether.

```
[bash]$ dig +nocmd @192.168.0.96 gotrice.com mx +noquestion +nostats
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 27349
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 3, ADDITIONAL: 5

;; ANSWER SECTION:
gotrice.com.            24000       IN    MX    10 mail.gotrice.com.
gotrice.com.            24000       IN    MX    20 drdre.gotrice.com.

;; AUTHORITY SECTION:
gotrice.com.            86400       IN    NS    shady.shadow.com.
gotrice.com.            86400       IN    NS    no.shadow.com.
gotrice.com.            86400       IN    NS    wutang.clan.org.

;; ADDITIONAL SECTION:
mail.gotrice.com.       24000       IN    A     192.168.0.32
drdre.gotrice.com.      24000       IN    A     192.168.0.33
shady.shadow.com.       86400       IN    A     192.168.0.96
no.shadow.com.          86400       IN    A     192.168.0.97
wutang.clan.org.        86400       IN    A     192.168.0.98
```

Similarly to nslookup, dig can also be used to query a specific type of DNS record, as in the above example, dig is used to query MX records for the domain gotrice.com. Evidently, dig by default offers a slightly bit more detailed output than nslookup which you can see in the Additional section, corresponding IP addresses of the mail servers are also provided.

As an additional note, if you still remember the five fields contained in a resource record responded by the DNS server, you should be able to interpret the output as shown above with no problem. For those who can't recall what those fields are, consider referring back to what has already been discussed in the `nslookup` section. However, let's try to read the first resource record provided in the Answer section of the above example, gotrice.com is the first field, representing for the domain name or owner of the records. The second field — 86400 — is the TTL (Time to Live) field, specifying the DNS server to remove the record from its cache after 86400 seconds. IN identifies the network class to be the Internet class. MX (Mail eXchanger) is the type of the record. The last field is of course the record data, which displays the data or values associated with the specified record type, as in this case, mail.gotrice.com.

Zone transfer is also possible by specifying `dig` to use the query type `axfr`:

```
[bash]$ dig +nocmd @192.168.0.96 gotrice.com axfr +noquestion +nostats

<results truncated for brevity>...

order.gotrice.com.      1D IN A         192.168.0.192
drdre.gotrice.com.      1D IN A         192.168.0.32
mail.gotrice.com.       1D IN A         192.168.0.33
db2.gotrice.com.        1D IN A         192.168.0.204
ttyl.gotrice.com.       1D IN CNAME     nagging
dephunk.gotrice.com.    1D IN A         192.168.0.69
                        1D IN HINFO     "1U-Rack" "Slackware"
                        1D IN MX 10     mail
                        1D IN NS        shady.shadow.com.
                        1D IN TXT       "Apache & MySQL"
dre.gotrice.com         1D IN A         192.168.0.21
                        1D IN HINFO     "admin" "win2k"
                        1D IN MX 20     drdre
                        1D IN TXT       "dre-owns-WarFTPd"
...<truncated>
.
```

Consider yourself very lucky if you are able to perform an unauthorized zone transfer successfully nowadays, since DNS administrators are becoming very security cautious; and thus, they are no longer playing Pac-Man while configuring the server. In practice, if you are not able to perform a zone transfer due to lack of authorization, you should change the query from type `axfr` to `any`, which will help you pull any records associated with the domain in question.

For extreme hackers who want to take control of the name server as well as the target network, the following query will help uncover version of the BIND name server:

```
[bash]$ dig +nocmd @192.168.0.96 version.bind chaos txt +noquestion \
+nostats

;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 27349
;; flags: qr aa rd ; QUERY: 1, ANSWER: 1 AUTHORITY: 0, ADDITIONAL: 0

;; ANSWER SECTION:
version.bind.           0           CH    TXT   "9.2.0"
```

Unfortunately, BIND version 9.2.0 is known to be vulnerable to a numerous Denial of Service (DoS) vulnerabilities, which hackers can exploit to effectively render the name service useless to legitimate users. Were the BIND version is 8.x and older; the situation could be even worse since the hackers may gain complete control of the vulnerable name server by exploiting any of a dozen buffer overflow vulnerabilities. Hackers who took control of the name server normally endeavor to take control of the target domain name for the purposes of blackmailing or defrauding the victim; however, this section won't delve too deep into hacking a DNS server since this is all about footprinting the target by using passive reconnaissance techniques. Some of you might argue about the word "passive" here because what has been discussed so far is using `dig` to make direct contact to the name server to collect information related to a particular target organization, which is rather aggressive. However, in the real world, the target normally does not have control of or own the DNS server; as a result, these reconnaissance attempts would most likely not be detected.

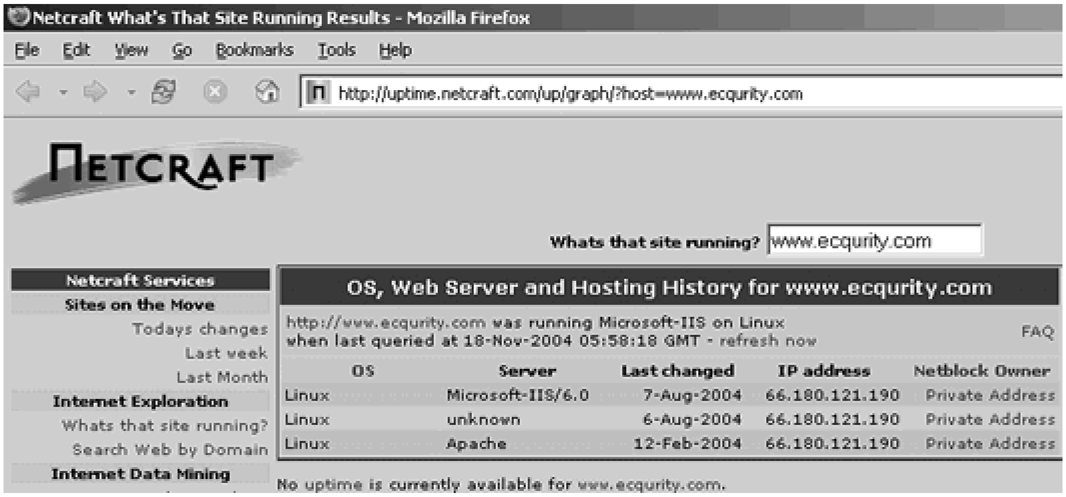
NETCRAFT

Netcraft is simply just too cool to be left out from any discussion of footprinting or information gathering. Established in 1995, Netcraft is an Internet service company based in Bath, England, providing a wide range of network security services, research data, and analysis on many aspects of the Internet. For more information regarding the company and their services, please feel free to visit http://www.netcraft.com.

Although Netcraft provides a wide range of Internet services but the one service which you should be looking at and checking out right now is "what's that site running?". The service box is located on the left frame of all Netcraft web pages, which means you can easily access it anytime you want, regardless of which page you're currently viewing. Using the service is rather simple since you only need to input the web address of the site in question to the box and click Search, Netcraft will then do the rest for you. The result will be varied depending on how much information Netcraft has about the site, but typically, Netcraft should be able to help you identify and detect the OS (Operating System), web server software, IP address, and network block owner of the site. Figure 2-7 shows you the results provided by Netcraft after a request was sent to discover what the host named www.ecqurity.com is running.

The results provided by Netcraft appears to be very straightforward, stating that the host http://www.ecqurity.com was running on Microsoft-IIS on Linux when was last queried at 18-Nov-2004 05:58:18 GMT. According to the results, you can see that Netcraft provided even more information than what I initially asked for. Not only result from my last query attempt was shown, but results of all previous queries cached by Netcraft were also shown in table format, separated by five different columns, representing for five different type of information related to the site.

In the first column, Netcraft offers information about the OS of www.ecqurity.com. Web server of the host is then identified and shown in the second column; note how three query attempts made against the site at three different times yielded three total different results, which I will explain why in a moment. The third column provides the dates since last change wherein the fourth one, the corresponding IP address of www.ecqurity.com is revealed. In the fifth column, if the host in question were part of a big corporate network, or belonged to a subnet, then information about the network block and its owner would also be shown, but since the host www.ecqurity.com is privately owned by me; therefore, the firth column shows nothing of interest.



Whats that site running? www.ecqurity.com

OS, Web Server and Hosting History for www.ecqurity.com

http://www.ecqurity.com was running Microsoft-IIS on Linux
when last queried at 18-Nov-2004 05:58:18 GMT - refresh now

OS	Server	Last changed	IP address	Netblock Owner
Linux	Microsoft-IIS/6.0	7-Aug-2004	66.180.121.190	Private Address
Linux	unknown	6-Aug-2004	66.180.121.190	Private Address
Linux	Apache	12-Feb-2004	66.180.121.190	Private Address

No uptime is currently available for www.ecqurity.com.

Figure 2-7. Netcraft - What's That Site Running

For those who are still wondering what sort of magic can make www.ecqurity.com run Microsoft IIS/6.0 smoothly on Linux, read the following answer provided by Netcraft in their FAQ page:

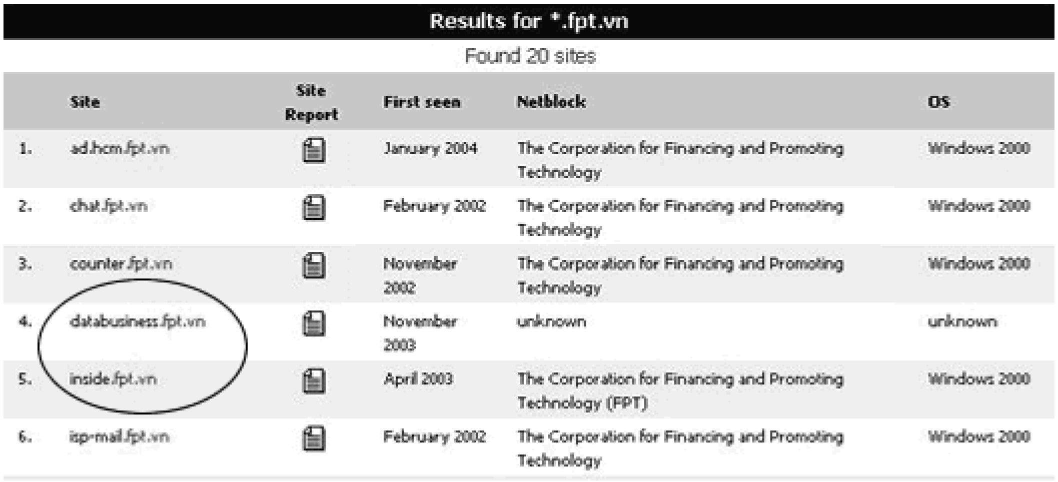
> *"Web servers that operate behind a caching system, load balancer, reverse proxy server or a firewall may sometimes report the operating system of the intermediate machine. Hence reports of 'Microsoft/IIS on Linux' may indicate that either the web server is behind a Linux server that is acting as a reverse proxy, or has configured the Akamai caching system such that the first request to the site goes to one of Akamai's servers [which run Linux], or as in the case of www.walmart.com has been configured to send a misleading signature."*

As you can read from the text above, the host www.ecqurity.com simply does not rely on Microsoft IIS/6.0 to run the show, for the reason that it "has been configured to send a misleading signature". My partner — David Coomber — purposely changed the banner of our web server so that it would appear to look more like a Microsoft IIS/6.0 server, while indeed, it is still the same-old-goodie Apache web server that runs on Linux. Note how Netcraft did return a proper and correct OS/web server combination when the query for the host www.ecqurity.com was first submitted on 12th/Feb/2004 and how subsequent queries made on the 6th and 7th Aug 2004 yielded different results due to changes in the web server's signature. From there, it is clear that Netcraft actually carries out live queries every time a request is made — rather than just using the cached results from previous queries — and the results provided by Netcraft, as you can see by yourself from Figure 2-7, are fairly reliable.

Ever since changing the signature of the web server, E-CQURITY has been receiving countless attacks that would have been possible if the OS platform and web server were indeed Microsoft Windows and IIS, respectively. Changing the signatures or banners of popular vulnerable network services is not a leading example or solution that would help you avoid being hacked, yet it is a good way to avoid some of the automate scanners or reconnaissance attempts from the script kiddies. Remember, security through obscurity is not always a good approach and an effective security solution does not merely rely on the "obscurity" part.

You may wonder why you should use Netcraft in lieu of some other sophisticated network reconnaissance tools out there. The answer is simple enough; you can get just as much information from Netcraft as from any other tools without having to expose and jeopardize your identity to the clients. Netcraft will stand in the middle, perform all your desired queries, and deliver the results back to you through the web interface. Compare that to the use of network tools, you will be exposing yourself to the target in which all reconnaissance attempts are made directly between you, the tools, and the target — assuming you do not use a proxy server or your proxy server is set to be transparent (which is not a truly anonymous configuration).

Besides telling you what the target operating system or web server is, Netcraft is also capable of searching for domain names. What this mean is that you enter a keyword of your choice into the domain search box and Netcraft will then attempt to search for any domain that contains the specified keyword. For instance, if you enter the exact domain of the target, without the lead www, such as .fpt.vn, into the search box, Netcraft will search for all occurrence sites that have domain names ending with .fpt.vn. There are many reasons behind why this type of searching is particularly useful. Firstly, searching for any domain names ending with .fpt.vn might help you unearth any secret or sensitive sites related to the network. Secondly, by finding all possible sites or domain names belong to the target, you might increase your chance of success since you now have more possible access points to the network. In Figure 2-8, Netcraft performed the query and returned 20 potential sites that have the domain names ending with .fpt.vn.



Results for *.fpt.vn

Found 20 sites

	Site	Site Report	First seen	Netblock	OS
1.	ad.hcm.fpt.vn		January 2004	The Corporation for Financing and Promoting Technology	Windows 2000
2.	chat.fpt.vn		February 2002	The Corporation for Financing and Promoting Technology	Windows 2000
3.	counter.fpt.vn		November 2002	The Corporation for Financing and Promoting Technology	Windows 2000
4.	databusiness.fpt.vn		November 2003	unknown	unknown
5.	inside.fpt.vn		April 2003	The Corporation for Financing and Promoting Technology (FPT)	Windows 2000
6.	isp-mail.fpt.vn		February 2002	The Corporation for Financing and Promoting Technology	Windows 2000

Figure 2-8. Netcraft - Search Web by Domain

Note the 4th and the 5th rows in the results; do they give you any hints at all? I won't go too much further into discussing what hackers normally do next since it is too obvious. However, to minimize the threats given by this type of reconnaissance attempt, it is imperative that under no circumstances must you give your sites attractive and informational names like *top-secret* or *inside*. Any site or domain named as such will just be a constant prey to countless probes and attacks given by hackers.

Unless the web-based network reconnaissance tools prove not to be as competent as the program-based tools, it is recommended that you should allow yourself to do all possible reconnaissance attempts through the portals like Netcraft, which will greatly increase the "safety" factor.

:.::.: ANALYZE AN EMAIL HEADER — TRACKING THE SENDER

The tool eMailTracker Pro as introduced earlier requires all users who intend to use the software for longer than the 15-day trial period must pay for a license fee, and this section is to show you how to save 30 bucks and not to pay for the software. No, don't get all so excited yet because I will not be showing you how to bypass the software protection or crack the software. Instead, you will learn how to obtain the same amount of information as the tool eMailTracker Pro by manually analyzing an email header, and it's just going to be as effective as using the software.

In order to obtain similar information as the tool eMailTracker providing, there are additional steps that you need to do, such as, performing DNS and network whois queries to verify the authenticity of the information provided in the email header as well as to identify the associated network block owner of the IP address in question.

Manually analyzing an email header is great for those who love to learn more about the inner details of how an email is transferred from one computer to another. Imagine being able to identify as much information about the target as possible just by analyzing a simple email from the target, wouldn't it be cool? A single email gives out more information about its sender than you initially thought, including internal and external IP address, geographical location, mail server's location and software version, mail client, host name of the sender's computer, date the message was composed, and so on. You might be wondering where in the world that information comes from, since all you can see from an email are the Subject, To, From, CC and BCC fields. Most of the mail clients, software-based or web-based, normally provide you a short header of the email for conciseness; thus, in order to acquire more information about the sender, you need to configure your email software to allow viewing of full email header, explicitly.

An example of a full email header is shown in the exhibit below:

Exhibit 2-5. A Full Email Header (Numbers added for readability)

```
1.   X-Apparently-To: swd@yahoo.com via 216.136.175.155; Mon, 26 Jul 2004 06:29:13 -0700
2.   X-Originating-IP: [208.252.123.45]
3.   Received: from 208.252.123.45 (EHLO mail.gotrice.com) (208.252.123.45) by
```

mta130.mail.dcn.yahoo.com with SMTP; Mon, 26 Jul 2004 06:29:13 -0700
4. Received: from drdre.gotrice.com ([192.168.0.251]) by mail.gotrice.com with Microsoft SMTPSVC(5.0.2195.6713); Mon, 26 Jul 2004 08:29:11 -0500
5. Subject: RE: Help
6. Date: Mon, 26 Jul 2004 08:29:11 -0500
7. Message-ID: <B3F6F52F53BBFE42B95CF2F96549183B0416C1@drdre.gotrice.com>
8. From: "Doctor Dre" <dr.dre@gotrice.com>
9. To: "Some Wicked Dude" <swd@yahoo.com>
10. Return-Path: Dr.Dre@gotrice.com
11. X-Mailer: Elephant 5.3

Obviously, there is a whole lot of information provided in an email header, which you can see from the above exhibit. However, the amount of information provided in an email header is not consistent as one time you might see it has a whole lot of information while at some other times you might only see a few lines. Generally, there are three players involved in a typical email transmission. The two players are of course the sender and receiver; after all, an email wouldn't be complete without the participation of the sender and receiver. The other yet important player that makes transferring email possible is the mail server. Note that there may be more machines or intermediary devices involved during the course of transferring an email, such as, firewall, Internet gateway, or proxy. But for now, let's just look at the three most common players that make up a successful email transmission. Figure 2-9 illustrates the role and position of these three players in a typical email transmission.

After an email message is composed, the mail client will add some basic information to the email header before handing it off to the mail server. You can see this from line 5 to line 11 in Exhibit 2-5 above. Line 5 describes the subject of the email message, rather self-explanatory. Line 6 indicates on which date and what time the message was composed. Line 7 is the Message-ID, encompassing a unique string assigned by the mail server for tracking and debugging purposes. The unique Message-ID also identifies from which machine the user logged in and sent the message, as in this example, you can see it was sent from the machine named drdre in the gotrice.com domain. Line 8 and 9 are the email addresses of the sender and receiver, drdre@gotrice.com and swd@yahoo.com, respectively. Line 10 specifies dr.dre@gotrice.com is the email address which all-return mails should go to. Line 11 indicating Elephant version 5.3 is the mail client that the sender used to compose and send the email message.

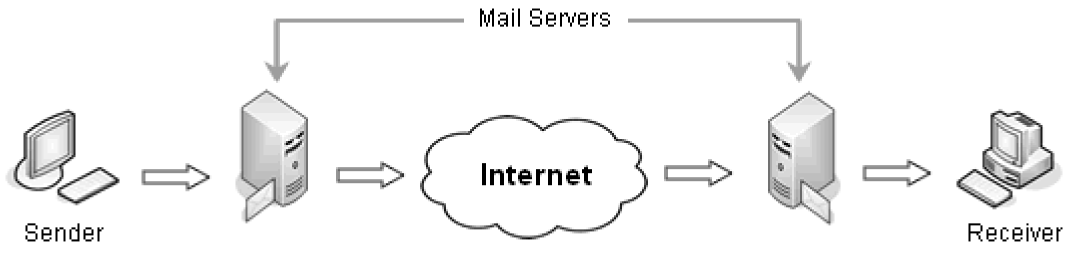


Figure 2-9. A typical email tranmission

The mail client once handed the message to the mail server will no longer have any control over the email. It is now the responsibility of the mail server to help deliver the message to its intended recipient by analyzing the message header to look for information about the recipient of the message, such as, the email address in the From or CC field. The sender's mail server will then have to add several more information into the message header before delivering the message to the recipient's mail server. Line 5 in the exhibit clearly illustrates this.

The Received field as in line 5 implies that on the 26th July 2004 at 08:29:11 Eastern Time, the mail server mail.gotrice.com receives a message originally handed off by the machine named drdre.gotrice.com whose corresponding IP address is 192.168.0.251; and yes, 192.168.0.251 is an internal IP address and it is non-routable. Additionally, the Received field also identifies the application and version information of mail.gotrice.com as Microsoft SMTP Service and 5.0.2195.6713. As soon as the mail server finishes adding its own header to the message, it will immediately deliver the message to where it should be — the receiver's mail server. As discussed above, the next destination after the first mail server is not always necessarily the receiver's mail server, it might be the Internet gateway, firewall, or any other filtering devices implemented at the sender's side.

Once the message arrived at the receiver's mail server, its header must be "abused" again for the last time by the mail server before it can actually reach its destination and stay safely in the receiver's mail inbox. Line 4 — the top Received field of the message header — is added by mta130.mail.dcn.yahoo.com, indicating that on 26th July 2004 at 06:29:13 Mountain Time, it receives an email message destined for user swd from the Internet domain mail.gotrice.com which has the associated IP address of 208.252.123.45.

Similarly to the approach that you take when analyzing `traceroute` output, it is best to analyze an email header from the bottom to the top, since the bottom reveals the most closely-related information associated with the sender of the email. However, it is worth noting that clever users can deliberately submit bogus information in an attempt to misdirect and cover their tracks; therefore, you need to perform additional verification steps, such as checking the geographical location of the originated IP address to make sure the email message comes from where you expect; and based on that IP address, performing reverse and forward DNS queries simultaneously.

Perhaps some of you are still wondering if this kind of information would be of any use in helping you hackers gain access to the target. It is indeed very useful. As you have seen earlier, a potentially large amount of information pertaining to the target network can be revealed by examining the full header of an email message that come from the target organization. The internal IP addressing information, the address of the mail server, the mail daemon application and its version and physical location of the server are what hackers love to obtain during this footprinting stage. In many cases, information about the proxy, firewall or gateway of the network is also revealed since it is very common for a network to have an email scanned before a client on the network can start sending or receiving email. And as usual, any time a message passed through a content-filtering device on the network, its header will be automatically added with information telling the others where and which device the message had gone through.

Obtaining an email and its full header from the target is not a hard task either, since there are many sources to look for such information. First of which, is on USENET or any particular mailing list that the target organization might be with. Alternatively, the hackers can also send an email to a bogus email address at the target; and thereby, get the fail delivery notification from the target's mail daemon. Everything sounds all nice and easy up until now, but what if the hackers could not obtain anything valuable in the email header? Well, that would lead them to the next footprinting trick, *web bug*.

:.::.: WEB BUG — TRACKING THE RECEIVER

Once upon a time, when email messages were just pure plain texts, in its original ASCII format, reading emails used to be an enjoyable moment and wasn't all that scary until now — the revolution of HTML-based email. The idea behind HTML email is to allow users using the conventions of HTML tags to format the email body; and thereby, make it look prettier and probably more readable. However, that seem-to-be-cool idea is also twisted all the way around by hackers and spammers, turning HTML email into a bug and an intrusive medium that assist them invade the privacy of the email recipient offensively. Receiving and reading HTML mails virtually become a living nightmare.

With the support of HTML tags in an email, hackers can easily plant a tiny and unnoticeable bug within the email and send them all together to the victim's email address. As soon as the victim receive and read the bugged mail, hackers will soon be able to find out the geographical location, IP address, date and time the email was read, the number of times that the email was read, how many times it's being forwarded, and a whole lot more. Sounds fun and easy, isn't it? In fact, there are many Internet companies providing this kind of service, which of course receives many criticisms from the public, especially the civil liberty group since the method used to track or spy on the email receiver is too intrusive. Nevertheless, the service is still growing unstoppable just because a gimmick like this makes sense to a lot of people *"Users of online dating services such as match.com who want to know if their potential dates are reading their messages...or ignoring them."* Well, who wouldn't fall for that?

The bug that has to be planted in an email can be a transparent gif image with 1x1 pixel or any kind of information that must be pulled from another server. This is certainly not a new technique as it has been around as long as the longevity of HTML-based email. The following is an example of a simple gif web bug concealed in an IMG tag within a HTML email:

```
<IMG SRC=”http://www.i-spy.com/img/bug.gif?ID=Mike>
```

Note how the image file bug.gif has an argument named ID followed by the specified keyword Mike; it is used for uniquely identifying the sent message. Only inserting the above line into a HTML email is not enough, the attackers need to have their own server that hosts the file bug.gif, as in this case, www.i-spy.com should be the web server belonged to the attackers. The moment the bugged mail is downloaded and viewed by the victim, the embedded IMG tag will fetch the file bug.gif from the www.i-spy.com web server, which in turn, leaving a footprint in the web server's log file. Hence, continue with the above example, after the bug.gif is pulled from the server, the line in the log file should look similar to something like this:

```
64.163.84.93 - - [13/Jul/2004:01:33:35 -0600] "GET /img/bug.gif?ID=MIKE
HTTP/1.0" 200 11874 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT
5.1; .NET CLR 1.1.4322)"
```

Evidently, the innocuous user, Mike, inadvertently provides the attackers with a lot of information merely by reading that bugged email. Tracing the IP address provided in the log, the attackers will be able to find out the location where Mike had viewed his email, geographically. Performing a simple Network whois query based on that IP address help the attackers reveal information about the ISP or network that Mike might be with. Moreover, the attackers now also know which OS platform as well as browser that Mike used while reading that email.

The given example above in no way can reflect the near level of severity or intrusiveness a web bug can induce. Yet it is a leading and simple example of showing what kind of information can be revealed by using a web bug. In practice, the spammers or attackers do not merely rely on the gif image, but they often use a script or dynamic web page to deliver the image instead; and by relying on the power of scripting, they can create a more intrusive and powerful bug that delivers more information about the victim. For instance, the attackers can craft a script that can time how long Mike has viewed the message by instructing the script to deliver the image at an unbelievably slow crawling speed, dribs and drabs style, to keep the connection opened as long as Mike is still reading. Once Mike is done with his reading, the connection will be closed, and the timer function of the script will also be stopped. The attackers then can use the information provided by the timer to deduce the reading time effectively. Many people have always believed that the longer time you take view the message the more serious you are about the sender or content of the message itself, but that has not always been the case since many people always prefer to read their emails offline.

As you have seen, even when there is no email header for the attackers to analyze and complete their footprinting of the target network, they can always use this web bug approach to do the job. Information revealed by using this sort of technique can vary depending on how creative the hackers can be with their bug as well as how the paranoid the victims can be about their privacy. In all, keep in mind that information about the OS, browser and IP address is very valuable to the attackers during footprinting.

The most important factor that makes this bug and homemade email-tracking system work is that mail clients of the recipients must have the capability to render IMG tag, or HTML tags collectively. With the HTML view feature turned off, the electronic bug would just be killed as good as if someone is spraying Flex directly at the little mosquito. Another yet important factor that makes mail-tracking works is that the recipients have to be connected to the Internet at the same time the bugged mail is viewed, otherwise the bug would have no way to contact the server. Therefore, it is important to keep in mind that most of privacy conscious users will have the loading HTML image feature disabled. Ironically, spammers and hackers are very creative in developing techniques to counter countermeasure, they have been known to use the IFRAME tag to track the recipients in lieu of the old web bug technique — a transparent gif image. But to keep thing simple, this section only gave you an overview of how to make a simple web bug with the help of the IMG tag. Discussion about advanced tracking methods, such as using the IFRAME tag, is out of the scope of this book, but for those who are interested to learn more about the tracking techniques as well as steps to prevent from being spied, consider paying the following sites a little visit:

[1] Building an Email Tracking System
http://www.wwwcoder.com/main/parentid/271/site/2313/68/default.aspx

[2] SecurityFocus: Securing Privacy Part 3: E-mail Issues
http://www.securityfocus.com/infocus/1579

[3] The Cookie Leak Security Hole in HTML Email Messages
http://www.computerbytesman.com/privacy/cookleak.htm

[4] Web Security, Privacy & Commerce, 2nd Edition: The Web's War on Your Privacy
http://www.oreilly.com/catalog/websec2/chapter/ch08.html