Information gathering: footprinting

License

Contents

General notes about the labs

Often the lab instructions are intentionally open ended, and you will have to figure some things out for yourselves. This module is designed to be challenging, as well as fun!

However, we aim to provide a well planned and fluent experience. If you notice any mistakes in the lab instructions or you feel some important information is missing, please feel free to add a comment to the document by highlighting the text and click the comment icon ( ), and I (Cliffe) will try to address any issues. Note that your comments are public.

If you notice others are also reading the lab document, you can click the chat icon ( ) to discuss the lab with each other.

Preparation

As with all of the labs in this module, start by loading the latest version of the LinuxZ template from the IMS system. If you have access to this lab sheet, you can read ahead while you wait for the image to load.

> To load the image: press F12 during startup (on the boot screen) to access the IMS system, then login to IMS using your university password. Load the template image: LinuxZ.

Once your LinuxZ image has loaded, log in using the username and password allocated to you by your tutor.

The root password -- **which should NOT be used to log in graphically** -- is "tiaspbiqe2r" (**t**his **i**s **a** **s**ecure **p**assword **b**ut **i**s **q**uite **e**asy **2** **r**emember). Again, never log in to the desktop environment using the root account -- that is bad practice, and should always be avoided.

Using the VM download script (as described in the previous lab), download these VMs:

- Kali Linux (Live disk or installed version). For the installed version use user:root password:toor

Feel free to read ahead while the VM is downloading.

Note that the lab instructions will probably not work outside the Leeds Met lab, since you would not have access to our example insecurely configured DNS server, so some of the information gathering attacks would not work.

Introduction to information gathering

Most attacks start at the information gathering stage. The aim is to gather as much information as possible about the target. The more information the attacker has, the more likely they will will be successful.

All sorts of information can be useful to an attacker. This process includes:

- gathering information that can be used in social engineering attacks
- detecting and identifying network ranges, and computers that are targets
- identifying all attack surface (systems and processes that are exposed to the attacker)
- finding services that are exposed, and finding out as much as possible about them
- querying services to see what other information can be gained, such as active usernames

Types of information gathering techniques

Information gathering techniques can be either:

- Passive: does not interact with any of the target's systems; there is no easy way to detect that this is happening
- Active: the attacker interacts with the target's systems: often in a hard to detect, or seemingly unobtrusive manner

Information gathering is sometimes further broken down to:

- Footprinting: mostly passive identification of network ranges and information about the target organisation
- Scanning: active phase, involving identifying IP addresses, ports, and services
- Enumeration: querying services for more information

Determining IP addresses

The first vital piece of information required to test the security of an organisation is a list of the IP addresses of the systems that are part of the organisation's network, and those that are exposed to the Internet.

An IP address is a numerical value that is used to identify a computer on a network, and is used in all communications over the Internet Protocol (IP), the main communication protocol used between computers, and which makes up the Internet. Because IP addresses can be hard for humans to remember, domain names are used. An example of a domain name is *leedsmet.ac.uk*.

The Domain Name System (DNS) is used to resolve a domain name to an IP address, so that the appropriate computer systems can communicate. This is similar to using a phone book to look up the number to call in order to contact a person. For example, when you open a Web browser and type “leedsmet.ac.uk”, the browser must first use DNS to resolve that to an IP address before it can actually start communicating with the Leeds Met web server.

Using dig

On the Kali Linux VM (the attacker), open a terminal by clicking the console icon.

Dig is a standard Unix command for conducting DNS queries.

Run:

```
dig +short example.com
```

The “+short” argument provides a terse answer. By default the DNS response will only includes “A records”: the IPv4 addresses of the servers that the name corresponds to.

Note that there can be multiple IP addresses associated with a single domain name.

To view the full verbose DNS response, run:

```
dig example.com
```

In addition to A records (IPv4 addresses), DNS contains other useful information. Search for information on the Internet, and read about these DNS record types:

- AAAA
- MX
- NS
- SOA

Run:

```
man dig
```

Note the synopsis:

```
SYNOPSIS
```

```
dig [@server] [-b address] [-c class] [-f filename] [-
k filename] [-m] [-p port#] [-q name] [-t type] [-x addr] [-
y [hmac:]name:key] [-4] [-6] [name] [type] [class]
[queryopt...]
```

This states that the command “dig” can followed by a number of different kinds of arguments. Following the name, such as “example.com”, we can specify the record types we are interested in.

Run:

```
dig example.com ANY
```

DNS queries start by contacting a DNS server (whichever the computer is configured to use, such as your ISP’s name server), and if that server does not know the answer it will ask another DNS server (and so on).

Use dig to do the following, and keep a record of the commands you use, and information you gather:

- determine the IPv4 address(es) for leedsmet.ac.uk
- determine the hostname(s) of the email server for ebay.co.uk (Hint: MX records)
- determine the IP address(es) for this hostname
- determine the name server(s) for google.com (Hint: NS records)

Reverse DNS lookup

Another useful DNS feature is the ability to do the opposite. Rather than then normal *forward lookup*, which starts with a hostname such as leedsmet.ac.uk and determines an IP address, a *reverse lookup* starts with an IP address and determines a hostname (based on the PTR record type). This can be extremely useful for security purposes.

The -x flag instructs dig to do a reverse lookup. Run:

```
dig -x 173.194.41.160
```

The results should show that the above IPv4 address is used to host google.com.

Run:

```
dig google.com
```

Because Google.com uses many IP addresses, this result may or may not include the above IP address.

Every IP address can be resolved to a name, even if it is not hosting a website.

Determine your Internet facing IP address, by visiting this website:

http://whatismyipaddress.com/

Use dig to do a reverse lookup on your IP address. For example, "dig -x *IPADDRESS*".

Determine an IP address for leedsmet.ac.uk, then do a reverse lookup on that IP address. Can you explain the result?

How would reverse lookups be useful to a security administrator that noticed an attack on their network from an IP address?

DNS zone transfers

A DNS zone transfer involves a DNS server responding with essentially all of the information in its database. This is one of the (older, but still used) methods used between DNS servers, so that nameservers can retrieve information from one another. From a security point of view, this data leakage could contain very useful information for an attacker, since it can be used by the attacker to form a network map of the organisation.

The Leeds Met IMS lab environment contains a DNS server vulnerable to zone transfers.

Tip: Outside of the lab environment, you could run these commands against zonetransfer.me rather than example.com, to perform a DNS zone transfer against a vulnerable name server. The zonetransfer.me domain is provided by Robin Wood, a local security consultant.

First determine the hostname of the local example.com DNS server...

Run:

```
dig example.com ANY
```

Note the name server in the response. This will likely start with "ns." or "ns1.". For example, "ns1.example.com".

A DNS zone transfer is triggered by specifying the “AXFR” record type for the DNS query.

Run:

```
dig @ns1.example.com example.com AXFR
```

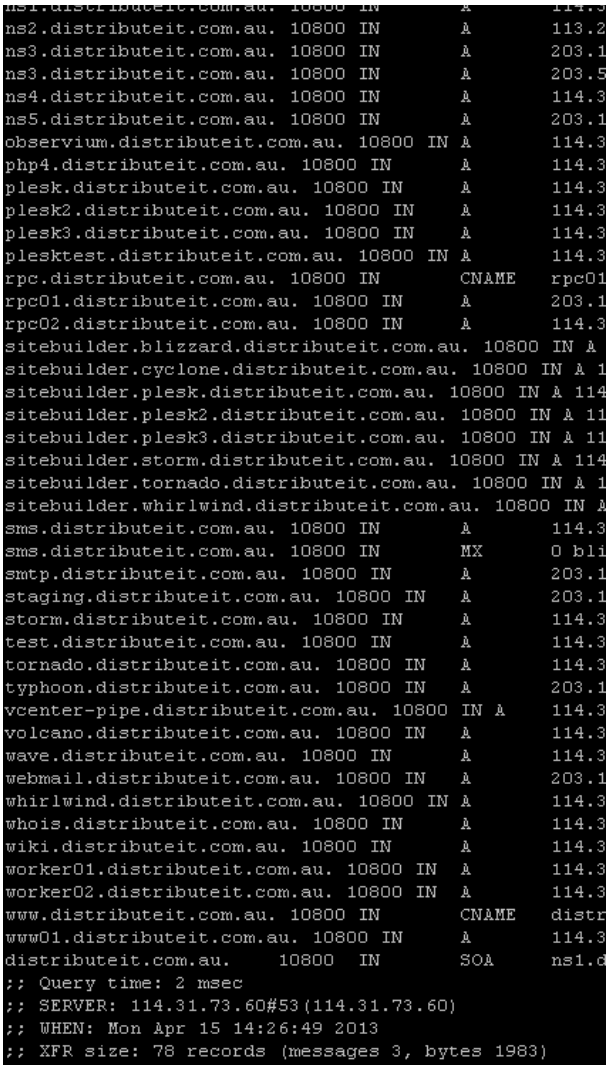
Where *ns1.example.com* is the name server noted previously.

If you conduct this from within the Leeds Met IMS lab environment, this “attack” should succeed.

If the information gathering attack succeeds, it will return with a list of all the IP addresses that the name server knows. These are IP addresses that an attacker can investigate in their search for vulnerable systems.

Save a copy of the output. How many IP addresses did you find?

DNS servers should be configured to only respond to other specific servers that it needs to get updates from: not an attacker’s IP address. However, sometimes DNS servers are misconfigured, and allow zone transfers. Below is a cropped example of a DNS zone transfer against a previously vulnerable *live* DNS server.



DNS zone transfer information gathering (IP addresses cropped from the right)

Other standard DNS tools

Most operating systems include DNS query tools, and dig is a common Unix command. Another Unix/Linux example is “host”.

Run:

```
host example.com
```

```
host google.com
```

Windows and Unix both have the “nslookup” command, although usage is different depending on the version.

Run:

```
nslookup example.com
```

Whois

When domain names are registered through a domain name registrar, the registrar records information such as contact details for the domain owner. This information is typically available via **Whois**.

Whois is the name of a command line tool, a TCP protocol, and the database containing the registration information.

From the command line “whois” can be used to obtain domain registration information, by querying the appropriate Whois servers. However, this will not work in our sandboxed labs.

As an example, ***outside our labs*** you could run “whois leedsmet.ac.uk”. The output is as follows:

```
[cliffe@cliffe-pc ~]$  whois leedsmet.ac.uk

Domain:
        leedsmet.ac.uk

Registered For:
        Leeds Metropolitan University

Domain Owner:
        Leeds Metropolitan University

Registered By:
        Jisc Collections and Janet Limited

Servers:
        dns0.lmu.ac.uk
        dns1.lmu.ac.uk
        dns2.lmu.ac.uk
        dns3.lmu.ac.uk


Registrant Contact:
        Tony Crossland

Registrant Address:
        Computing Services
        Room C717
        City Campus
```

```
                Calverley Street
                Leeds
                LS1 3HE
                United Kingdom
                +44 113 2833119 (Phone)
                +44 113 2835962 (FAX)
                t.crossland@lmu.ac.uk

        Renewal date:
                Tuesday 14th Apr 2015

        Entry updated:
                Wednesday 20th March 2013

        Entry created:
                Wednesday 17th September 2003
```

The above output contains a contact person, an email address, phone number, and a physical address; all of which are potentially useful to an attacker.

Consider this: how could each of these pieces of information be useful to an attacker?

Visit a website that provides whois lookups (there are many available), and search for information about leedsmet.ac.uk, linux.com, and then google.com.

Example websites include:

- whois.com
- who.is
- networksolutions.com/whois

Note that some registrars offer a privacy service, which typically costs money. This usually results in the whois database containing information about how to contact the registrar rather than the domain owner, which is less useful to an attacker.

Lookup the whois information of a local small business (or individual) with a website. How could the exposed information could be used in a social engineering attack?

Another important feature of Whois, is the ability to start from an IP address, and not only obtain information about the registered domain name owner, but also the range of IP addresses that they have.

If we know any single IP address belonging to an organisation, such as Leeds Met, then we can determine the full range of IP addresses that have been allocated.

So for example, once we have an IP address using dig...

```
[cliffe@cliffe-pc ~]$  dig leedsmet.ac.uk +short A
160.9.244.59
160.9.134.58
160.9.134.59
160.9.244.58
```

Then we can use the whois command on that IP address... (Remember, in the IMS labs you need to use a Whois website rather than the command)

```
[cliffe@cliffe-pc ~]$  whois 160.9.134.58

#
# ARIN WHOIS data and services are subject to the Terms of Use
# available at: https://www.arin.net/whois_tou.html
#


#
# Query terms are ambiguous.  The query is assumed to be:
#      "n 160.9.134.58"
#
# Use "?" to get help.
#

#
# The following results may also be obtained via:
#
http://whois.arin.net/rest/nets;q=160.9.134.58?
showDetails=true&showARIN=false&ext=netref2
#

NetRange:       160.8.0.0 - 160.9.255.255
CIDR:           160.8.0.0/15
OriginAS:
NetName:        RIPE-ERX-160-8-0-0
NetHandle:      NET-160-8-0-0-1
Parent:         NET-160-0-0-0-0
NetType:        Early Registrations, Transferred to RIPE NCC
Comment:        These addresses have been further assigned to users in
Comment:        the RIPE NCC region.  Contact information can be found in
Comment:        the RIPE database at http://www.ripe.net/whois
RegDate:        2004-04-05
Updated:        2004-04-05
Ref:            http://whois.arin.net/rest/net/NET-160-8-0-0-1

OrgName:        RIPE Network Coordination Centre
OrgId:          RIPE
Address:        P.O. Box 10096
City:           Amsterdam
StateProv:
PostalCode:     1001EB
```

```
Country:       NL
RegDate:
Updated:       2011-09-24
Ref:           http://whois.arin.net/rest/org/RIPE

ReferralServer: whois://whois.ripe.net:43

OrgTechHandle: RNO29-ARIN
OrgTechName:   RIPE NCC Operations
OrgTechPhone:  +31 20 535 4444
OrgTechEmail:  ncc@ripe.net
OrgTechRef:    http://whois.arin.net/rest/poc/RNO29-ARIN

OrgAbuseHandle: RNO29-ARIN
OrgAbuseName:   RIPE NCC Operations
OrgAbusePhone:  +31 20 535 4444
OrgAbuseEmail:  ncc@ripe.net
OrgAbuseRef:   http://whois.arin.net/rest/poc/RNO29-ARIN


#
# ARIN WHOIS data and services are subject to the Terms of Use
# available at: https://www.arin.net/whois_tou.html
#



Found a referral to whois.ripe.net:43.

% This is the RIPE Database query service.
% The objects are in RPSL format.
%
% The RIPE Database is subject to Terms and Conditions.
% See http://www.ripe.net/db/support/db-terms-conditions.pdf

% Note: this output has been filtered.
%       To receive output for a database update, use the "-B" flag.

% Information related to '160.9.0.0 - 160.9.255.255'


inetnum:       160.9.0.0 - 160.9.255.255
netname:       LEEDSMET-NET
descr:         Leeds Metropolitan University
descr:         Calverley Street
descr:         Leeds
descr:         LS1 3HE
country:       GB
admin-c:       CJE5-RIPE
tech-c:        CJE5-RIPE
mnt-by:        MNT-LEEDSMET
status:        early-registration
source:        RIPE # Filtered

person:        Chris Evans
address:       Computing Services
address:       Leeds Metropolitan University
address:       Calverley Street
address:       Leeds
```

```
address:        LS1 3HE
phone:          +441132833101
fax-no:         +441132833145
nic-hdl:        CJE5-RIPE
source:         RIPE # Filtered

% Information related to '160.9.0.0/16AS786'

route:          160.9.0.0/16
descr:          Leeds Metropolitan University
descr:          Calverley Street
descr:          Leeds LS1 3HE
descr:          UNITED KINGDOM
origin:         AS786
mnt-by:         JIPS-NOSC
source:         RIPE # Filtered

% This query was served by the RIPE Database Query Service version 1.66.3 (WHOIS4)
```

The output above includes the fact that the IP address belongs to Leeds Met, and also that Leeds Met has IPv4 address block of 160.8.0.0 - 160.9.255.255, which is a huge IP address allocation.

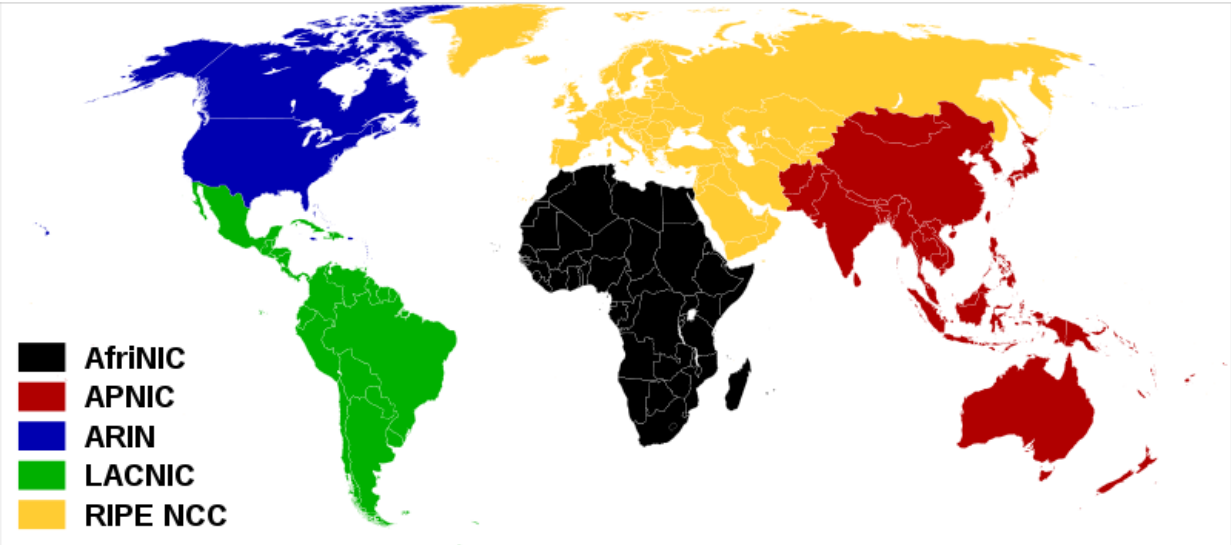
An attacker could use this information to direct their next stages of investigation.

Determine the IP address range used by Redhat (redhat.com).

Regional Internet Registries

Whois servers are operated by Regional Internet Registries (RIR), each of which are responsible for data for corresponding regions, as illustrated below.

Region	RIR
Africa	AfriNIC
Asia and the Pacific	APNIC
North America	ARIN
Latin America	LACNIC
Europe	RIPE



Regional Internet Registries and their catchment areas (by Canuckguy et al. licensed under the Creative Commons Attribution-Share Alike 3.0 Unported license)

DNS information gathering tools for penetration testing

Kali Linux includes many DNS information gathering tools, many specifically designed for security testing.

Use **dnstracer** to see how some domain names are resolved, and which DNS servers are queried. Run:

```
dnstracer example.com
```

```
dnstracer leedsmet.ac.uk
```

Note: this can take some time, so you may want to continue on in another Konsole tab (Ctrl-Shift-T).

Subdomain brute-forcing

Many organisation's domain names also contain subdomains, which point to other IP addresses. DNS brute-forcing involves guessing the likely names, to detect the IP addresses of other systems on the network. For example, it would make sense to try **mail**.google.com or **mx**.google.com to see if they point at anything.

Software exists that can automate this process, such as **Dnsmap**. Run:

```
dnsmap example.com
```

Dnsmap has a built in list of subdomains that it attempts to resolve, and alternatively you can download a larger list to use. Within the lab, the output from the above command should detect some subnets.

DNSenum and **Fierce** are security testing tools that attempt to gather as much DNS information about domains as they can.

Some of the things these tools attempt are DNS lookups, zone transfers, bruteforcing subdomains, gathering information from Google searches, and performing whois lookups.

Run each of these pen testing tools against example.com.

For example, run each of the following:

```
dnsenum example.com
```

```
fierce -dns example.com
```

After each, take the time to understand the output. The output gives the ethical hacker lots of information about the network, including the result of zone transfers and standard DNS queries.

You are encouraged to experiment with command line options. For a list of options:

```
dnsenum -h
```

Dnsrecon can also perform many of these kinds of security tests. For example, attempt a DNS zone transfer:

```
dnsrecon.py -t axfr -d example.com
```

It is not unusual for security testers and companies to write their own automation scripts, based around what suits their own testing methodology. For example, the Leeds-based network security solutions company Sec-1 use their own script known as “interrogator”, which performs these kinds of tests.

Conclusion

At this point you have:

- Learned about DNS and used related Linux commands to determine IP addresses and domains associated with IP addresses
- Used Whois to fetch information about domain ownership and IP ranges

- Understood and launched a DNS zone transfer “attack” against a misconfigured DNS server
- Used various automated DNS footprinting tools

Well done! In many cases, this provides enough information to move on to the scanning phase of an attack on a target, which we will cover next.