# Lab3 Recon-ng Lab

## Due by midnight February 2 (Thr), 2023

### Lab Learning Objectives

- Perform DNS recon using commands such as host and nslookup
- Use Recon-ng to perform DNS reverse lookup
- Use Recon-ng to perform DNS cache snooping to identify antivirus tools used by the target organizations

### Lab Setup

In this lab, you will use the Kali Linux virtual machine.

### Lab Instructions

In this lab, you'll gain familiarity with the user interface of Recon-ng and use it to gather useful information from the target organization's DNS infrastructure.

You'll run a Recon-ng module called reverse-resolve, which takes a netblock of IP addresses and sends PTR (reverse record) lookups to a DNS server to determine which of those IP addresses resolve into names. That's a useful feature for a penetration tester, because it can help you identify hosts that could be included in your scope, provided that these hosts have PTR records in DNS. Many organizations provide PTR records for important hosts on the internet, so this technique can be helpful during the reconnaissance phase.

Next, you'll use a Recon-ng module called "cache-snoop" that performs DNS cache snooping against a target DNS server. This module looks for cached DNS records associated with a couple dozen antivirus firms' signature update site DNS records. By identifying those cached entries, a penetration tester can determine which antivirus products the target organization is using, a helpful piece of information useful in evading the organization's AV product.

0. Start the lab by making sure you are running with root privileges (with a # prompt). You can achieve that by typing

```
$ sudo su -
```

First we will gather information for UCM's DNS server. Bring up a terminal and run the host command to obtain the DNS server

```
# host -t ns ucmo.edu
```

We can see that ns2.ucmo.edu is UCM's primary DNS server (ns1.ucmo.edu is secondary). Next, we need to know the IP address for ns2.ucmo.edu. We can run;

```
# nslookup ns2.ucmo.edu
```

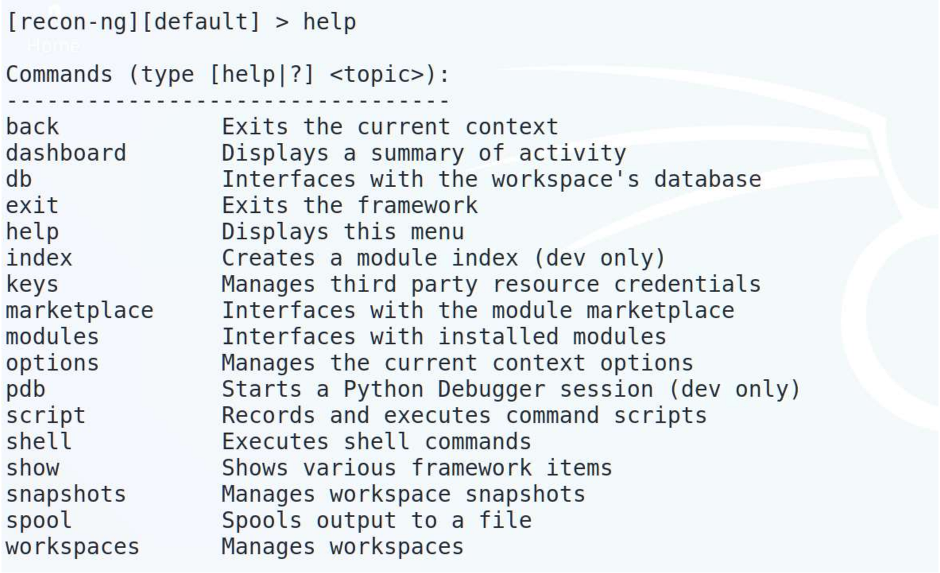
We find that the IP address for ns2.ucmo.edu is 153.91.1.52.

1. Start Recon-ng by bring up another terminal and type

```
# recon-ng
```

2. Recon-ng 5 comes without modules, which is also one of the major differences from the previous versions. To become familiar with Recon-ng's user interface, let's explore its help feature

**[recon-ng] > help**



```
[recon-ng][default] > help

Commands (type [help|?] <topic>):
---------------------------------
back            Exits the current context
dashboard       Displays a summary of activity
db              Interfaces with the workspace's database
exit            Exits the framework
help            Displays this menu
index           Creates a module index (dev only)
keys            Manages third party resource credentials
marketplace     Interfaces with the module marketplace
modules         Interfaces with installed modules
options         Manages the current context options
pdb             Starts a Python Debugger session (dev only)
script          Records and executes command scripts
shell           Executes shell commands
show            Shows various framework items
snapshots       Manages workspace snapshots
spool           Spools output to a file
workspaces      Manages workspaces
```

3. To see the variables set in Recon-ng, run

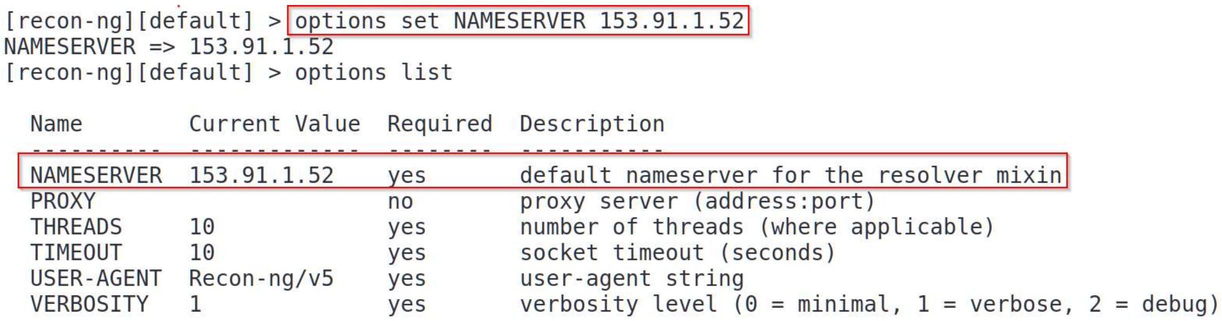
**[recon-ng] > options list**

Here, you can see that by default, Recon-ng resolves information using the 8.8.8.8 name server provided by Google. We'll change that shortly to UCM's DNS server by typing

**[recon-ng] > options set NAMESERVER 153.91.1.52**

Now, when we run options list, we can see that the original 8.8.8.8 name server has been altered to 153.91.1.52.

**[recon-ng] > options list**



```
[recon-ng][default] > options set NAMESERVER 153.91.1.52
NAMESERVER => 153.91.1.52
[recon-ng][default] > options list

  Name        Current Value  Required  Description
  ----------  -------------  --------  -----------
  NAMESERVER  153.91.1.52    yes       default nameserver for the resolver mixin
  PROXY                      no        proxy server (address:port)
  THREADS     10             yes       number of threads (where applicable)
  TIMEOUT     10             yes       socket timeout (seconds)
  USER-AGENT  Recon-ng/v5    yes       user-agent string
  VERBOSITY   1              yes       verbosity level (0 = minimal, 1 = verbose, 2 = debug)
```

4. Let's now explore the various modules Recon-ng has. Since version 5 no modules are available by default, we add them using the command marketplace. But first, the module list should be updated with the command marketplace refresh.

**[recon-ng] > marketplace refresh**

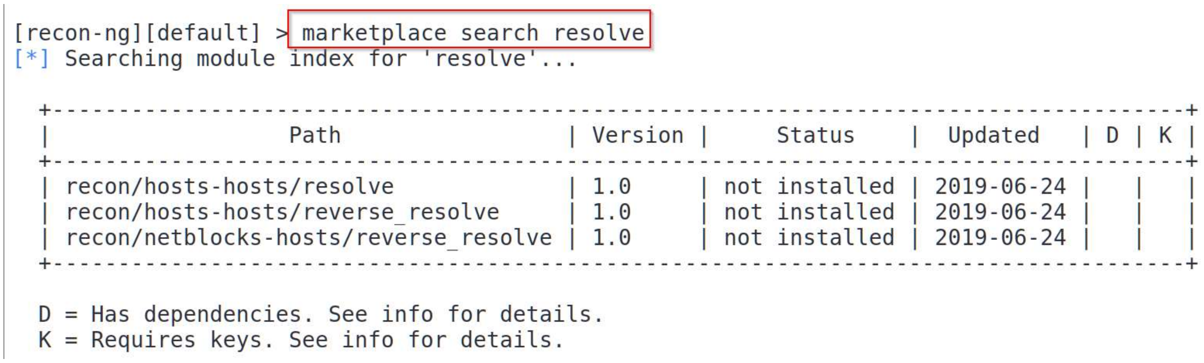
5. To find all available modules, type

**[recon-ng] > marketplace search**

We can also use the search command to find specific modules based on strings in the module's name or path. Suppose, for example, we wanted to find modules that would resolve names (via either a forward or a reverse DNS lookup). We could simply run search resolve.

**[recon-ng] > marketplace search resolve**

Here, we can see several modules associated with resolving names. Notice that their paths all start with recon, as they are in the recon module group.



```
[recon-ng][default] > marketplace search resolve
[*] Searching module index for 'resolve'...

  +---------------------------------------------------------------------------------+
  |                  Path                  | Version |     Status    |  Updated   | D | K |
  +---------------------------------------------------------------------------------+
  | recon/hosts-hosts/resolve              | 1.0     | not installed | 2019-06-24 |   |   |
  | recon/hosts-hosts/reverse_resolve      | 1.0     | not installed | 2019-06-24 |   |   |
  | recon/netblocks-hosts/reverse_resolve  | 1.0     | not installed | 2019-06-24 |   |   |
  +---------------------------------------------------------------------------------+

  D = Has dependencies. See info for details.
  K = Requires keys. See info for details.
```

To install all modules, type

**[recon-ng] > marketplace install all**

You may notice that a few modules require api key to run (ignore them for now).

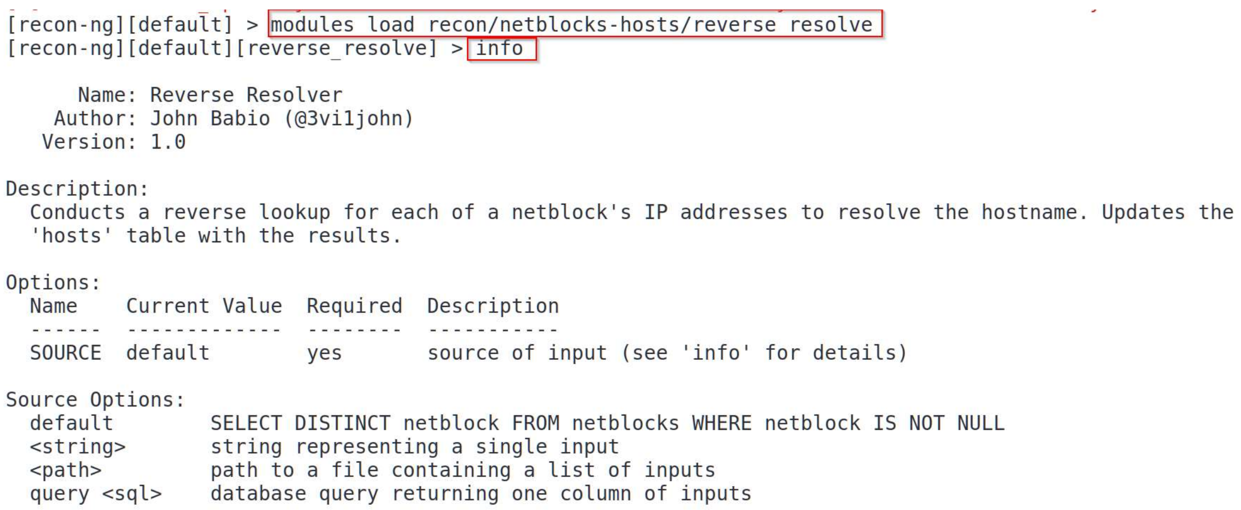
6. For this lab, we'd like to iterate through a given netblock (153.91.153.0/24) to see which host IP addresses have an associated PTR record. This is a useful way to find hosts and explore our scope in a penetration test. Of course, not every host on the internet has a PTR record, but many DMZ systems do, and we can use this module to help identify them.

Let's select that recon/netblocks-hosts/reverse_resolve module with the use command, followed by the full path to the module

**[recon-ng] > modules load recon/netblocks-hosts/reverse_resolve**

Now, to get the details of that module, we can run

**[recon-ng] > info**



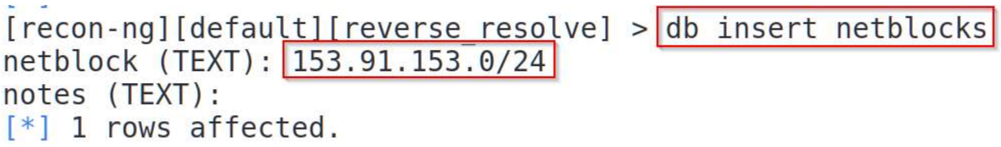
```
[recon-ng][default] > modules load recon/netblocks-hosts/reverse_resolve
[recon-ng][default][reverse_resolve] > info

      Name: Reverse Resolver
    Author: John Babio (@3vi1john)
   Version: 1.0

Description:
  Conducts a reverse lookup for each of a netblock's IP addresses to resolve the hostname. Updates the
  'hosts' table with the results.

Options:
  Name    Current Value  Required  Description
  ------  -------------  --------  -----------
  SOURCE  default        yes       source of input (see 'info' for details)

Source Options:
  default        SELECT DISTINCT netblock FROM netblocks WHERE netblock IS NOT NULL
  <string>       string representing a single input
  <path>         path to a file containing a list of inputs
  query <sql>    database query returning one column of inputs
```

For this module, the SOURCE variable specifies where the information about our target netblock comes from. By default, Recon-ng simply looks in the netblocks table. Currently, there is no data stored in the netblocks table. You can verify this by typing

**[recon-ng] > show netblocks**

7. Let's add a netblock 153.91.153.0/24 to the netblocks table using the add command:

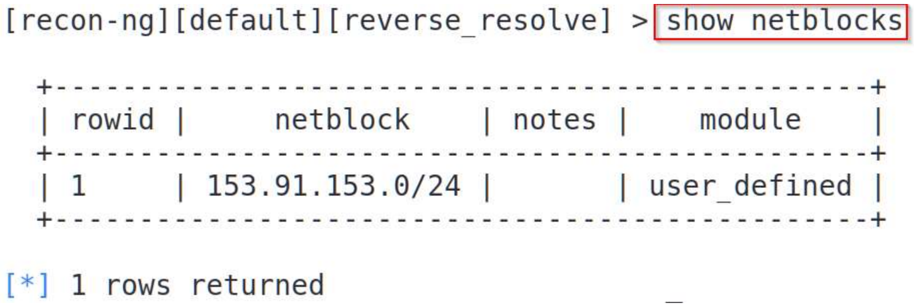
**[recon-ng] > db insert netblocks**



```
[recon-ng][default][reverse_resolve] > db insert netblocks
netblock (TEXT): 153.91.153.0/24
notes (TEXT):
[*] 1 rows affected.
```

We can now look at the netblocks table to see our information

**[recon-ng] > show netblocks**



```
[recon-ng][default][reverse_resolve] > show netblocks

  +-------------------------------------------+
  | rowid |     netblock    | notes |   module     |
  +-------------------------------------------+
  | 1     | 153.91.153.0/24 |       | user_defined |
  +-------------------------------------------+

[*] 1 rows returned
```

Alternatively, you can skip step 7 by directly setting the netblocks using the SOURCE option

**[recon-ng] > options set SOURCE 153.91.153.0/24**

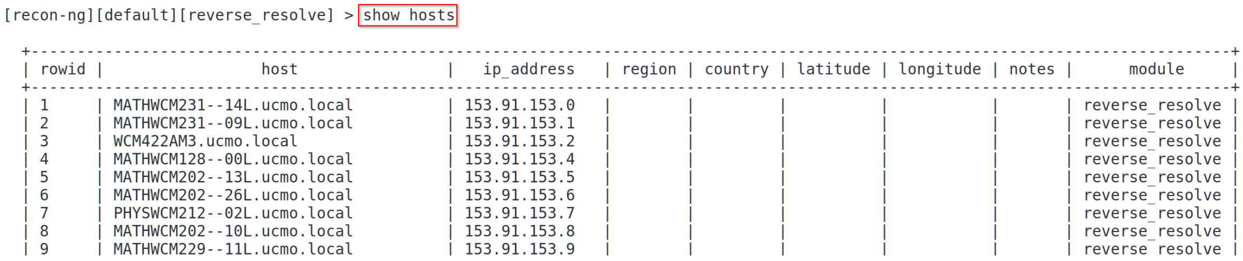
Make sure that the option name must be typed in in all caps.

8. With our module configured, we can now run it as follows:

**[recon-ng] > run**

9. In addition to scrolling back on the screen to see what Recon-ng found, we can also look at the hosts table, because the reverse_resolve module automatically populates it. Let's look at our newly discovered hosts

**[recon-ng] > show hosts**



```
[recon-ng][default][reverse_resolve] > show hosts

  +----------------------------------------------------------------------------------------------------------------------------------+
  | rowid |                host                |   ip_address    | region | country | latitude | longitude | notes |      module     |
  +----------------------------------------------------------------------------------------------------------------------------------+
  | 1     | MATHWCM231--14L.ucmo.local         | 153.91.153.0    |        |         |          |           |       | reverse_resolve |
  | 2     | MATHWCM231--09L.ucmo.local         | 153.91.153.1    |        |         |          |           |       | reverse_resolve |
  | 3     | WCM422AM3.ucmo.local               | 153.91.153.2    |        |         |          |           |       | reverse_resolve |
  | 4     | MATHWCM128--00L.ucmo.local         | 153.91.153.4    |        |         |          |           |       | reverse_resolve |
  | 5     | MATHWCM202--13L.ucmo.local         | 153.91.153.5    |        |         |          |           |       | reverse_resolve |
  | 6     | MATHWCM202--26L.ucmo.local         | 153.91.153.6    |        |         |          |           |       | reverse_resolve |
  | 7     | PHYSWCM212--02L.ucmo.local         | 153.91.153.7    |        |         |          |           |       | reverse_resolve |
  | 8     | MATHWCM202--10L.ucmo.local         | 153.91.153.8    |        |         |          |           |       | reverse_resolve |
  | 9     | MATHWCM229--11L.ucmo.local         | 153.91.153.9    |        |         |          |           |       | reverse_resolve |
```

10. The second part of the lab will use another Recon-ng module to determine the most likely antivirus tool or tools UCM is using from DNS cache. We can do that with the cache_snoop module in Recon-ng's discovery group. We can back out of our current module to the general Recon-ng prompt using the back command

**[recon-ng] > back**

We'll now use the discovery/info_disclosure/cache_snoop module

**[recon-ng] > modules load discovery/info_disclosure/cache_snoop**

Let's look at the options for this module

**[recon-ng] > info**

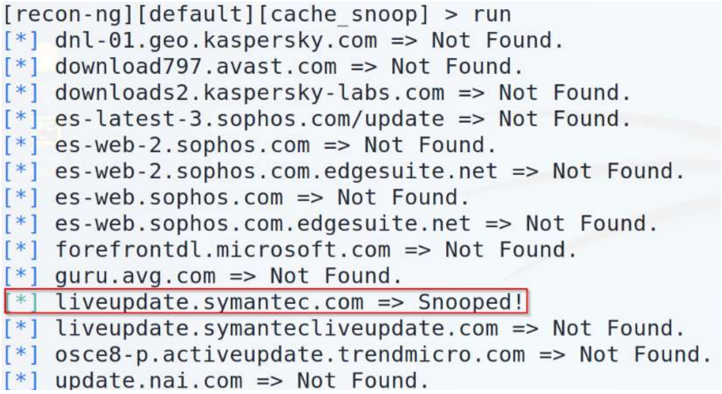
Here, we see that this module needs a NAMESERVER. Setting the name server by typing

**[recon-ng] > options set NAMESERVER 153.91.1.52**

With all our settings now in place, we can run the module

**[recon-ng] > run**

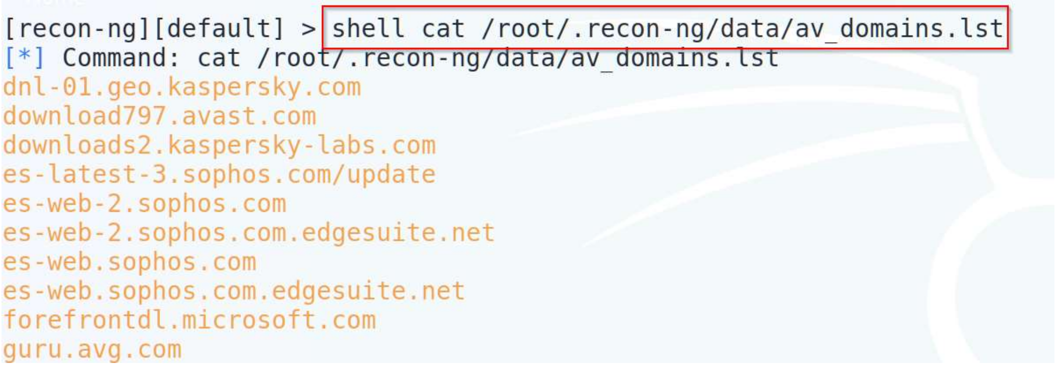
As the module runs, look carefully at its output. You'll note that it says that UCM uses symantec. Based on the cache contents, you may not get any finding. It is worth noting you may see different output as the university may change its anti-virus product.



```
[recon-ng][default][cache_snoop] > run
[*] dnl-01.geo.kaspersky.com => Not Found.
[*] download797.avast.com => Not Found.
[*] downloads2.kaspersky-labs.com => Not Found.
[*] es-latest-3.sophos.com/update => Not Found.
[*] es-web-2.sophos.com => Not Found.
[*] es-web-2.sophos.com.edgesuite.net => Not Found.
[*] es-web.sophos.com => Not Found.
[*] es-web.sophos.com.edgesuite.net => Not Found.
[*] forefrontdl.microsoft.com => Not Found.
[*] guru.avg.com => Not Found.
[*] liveupdate.symantec.com => Snooped!
[*] liveupdate.symantecliveupdate.com => Not Found.
[*] osce8-p.activeupdate.trendmicro.com => Not Found.
[*] update.nai.com => Not Found.
```

The names of many popular anti-virus products update servers are included in the av_domains.lst file that comes with Recon-ng. We can look at the contents of that file by typing

**[recon-ng] > shell cat /root/.recon-ng/data/av_domains.lst**



```
[recon-ng][default] > shell cat /root/.recon-ng/data/av_domains.lst
[*] Command: cat /root/.recon-ng/data/av_domains.lst
dnl-01.geo.kaspersky.com
download797.avast.com
downloads2.kaspersky-labs.com
es-latest-3.sophos.com/update
es-web-2.sophos.com
es-web-2.sophos.com.edgesuite.net
es-web.sophos.com
es-web.sophos.com.edgesuite.net
forefrontdl.microsoft.com
guru.avg.com
```

We can see the names of update servers for numerous antivirus product companies. You could expand this list based on different items you'd like to snoop for in a target organization's DNS cache. This information about the target's AV vendor is tremendously useful in our penetration test, especially if we are going to create any malware for the target organization to send via spear phishing or other means. It is worth noting that cat is not a recon-ng command. In order to run a shell command at the recon-ng prompt, start the command using shell. When Recon-ng receives a command it does not recognize, it passes that command to the underlying operating system shell for execution. This is handy because it means that we can run general purpose commands at the recon-ng prompt.

12. Next, let's use Recon-ng to find some interesting files on the UCM website by using the **discovery/info_disclosure/interesting_files** module.

**[recon-ng] > modules load discovery/info_disclosure/interesting_files**

Let's get some info on this module by typing

**[recon-ng] > info**

There are a few options we need to set

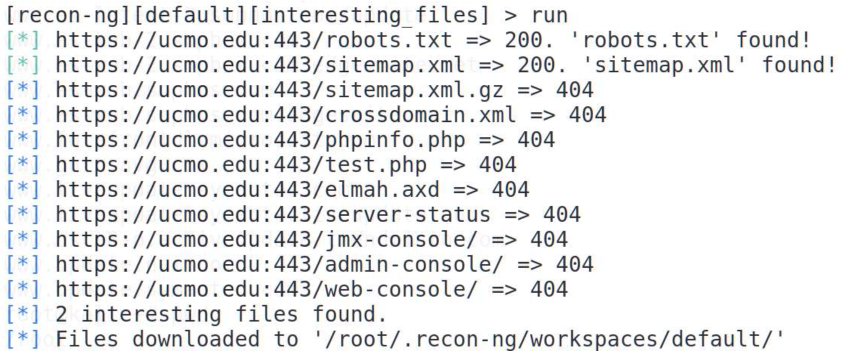
**[recon-ng] > options set SOURCE ucmo.edu**

**[recon-ng] > options set PROTOCOL https**

**[recon-ng] > options set PORT 443**

With all our settings now in place, we can run the module

**[recon-ng] > run**



```
[recon-ng][default][interesting_files] > run
[*] https://ucmo.edu:443/robots.txt => 200. 'robots.txt' found!
[*] https://ucmo.edu:443/sitemap.xml => 200. 'sitemap.xml' found!
[*] https://ucmo.edu:443/sitemap.xml.gz => 404
[*] https://ucmo.edu:443/crossdomain.xml => 404
[*] https://ucmo.edu:443/phpinfo.php => 404
[*] https://ucmo.edu:443/test.php => 404
[*] https://ucmo.edu:443/elmah.axd => 404
[*] https://ucmo.edu:443/server-status => 404
[*] https://ucmo.edu:443/jmx-console/ => 404
[*] https://ucmo.edu:443/admin-console/ => 404
[*] https://ucmo.edu:443/web-console/ => 404
[*] 2 interesting files found.
[*] Files downloaded to '/root/.recon-ng/workspaces/default/'
```

Recon-ng finds two interesting files from the UCM website. You can review the contents of those files by typing

www.ucmo.edu/robots.txt

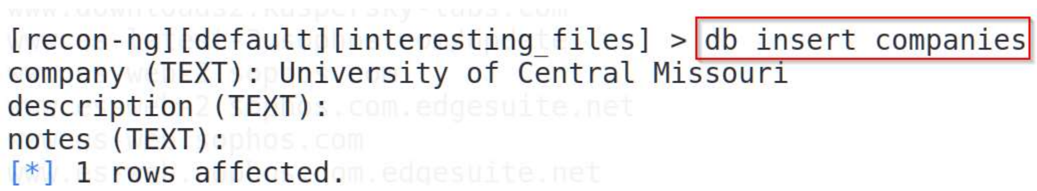
www.ucmo.edu/sitemap.xml

13. Next we will use the **recon/domains-hosts/google_site_web** module to find some hosts belong to ucmo.edu domain. In order to perform a search on the targeted domain, we first need to add the targeted domain to our database. Next, we will add our company and our domain. This will add information to the SQLite database with recon-ng. To add information into the database, we need to understand the schema, the layout of the tables. To look at the schema of the database run the following command

**[recon-ng]> db schema**

From the output, you can find 13 tables in the database. We will use the companies and domains tables for our task. Type the following to add University of Central Missouri to the companies table. Press enter if you want to leave some entries blank.

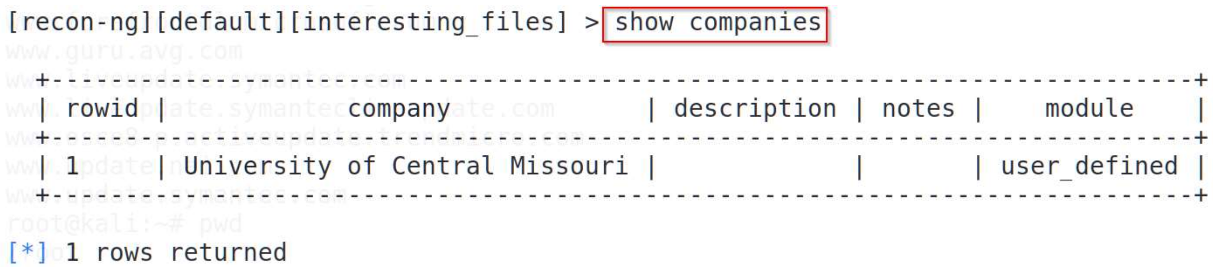
**[recon-ng][UCM]> db insert companies**



```
[recon-ng][default][interesting_files] > db insert companies
company (TEXT): University of Central Missouri
description (TEXT):
notes (TEXT):
[*] 1 rows affected.
```

To verify that the companies table got modified, type

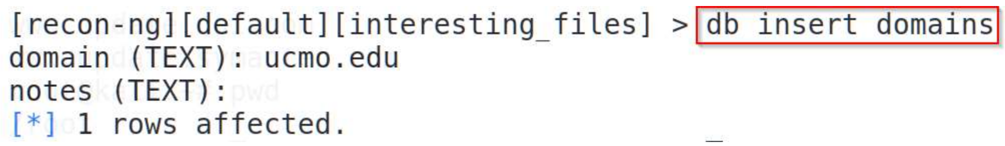
**[recon-ng][UCM]> show companies**



```
[recon-ng][default][interesting_files] > show companies

  +------------------------------------------------------------------------------+
  | rowid |            company             | description | notes |    module    |
  +------------------------------------------------------------------------------+
  | 1     | University of Central Missouri |             |       | user_defined |
  +------------------------------------------------------------------------------+

[*] 1 rows returned
```

We will now add the domain **ucmo.edu** to the domains table. Type

**[recon-ng][UCM]> db insert domains**



```
[recon-ng][default][interesting_files] > db insert domains
domain (TEXT): ucmo.edu
notes (TEXT):
[*] 1 rows affected.
```

Again, you can verify the newly added domain by typing

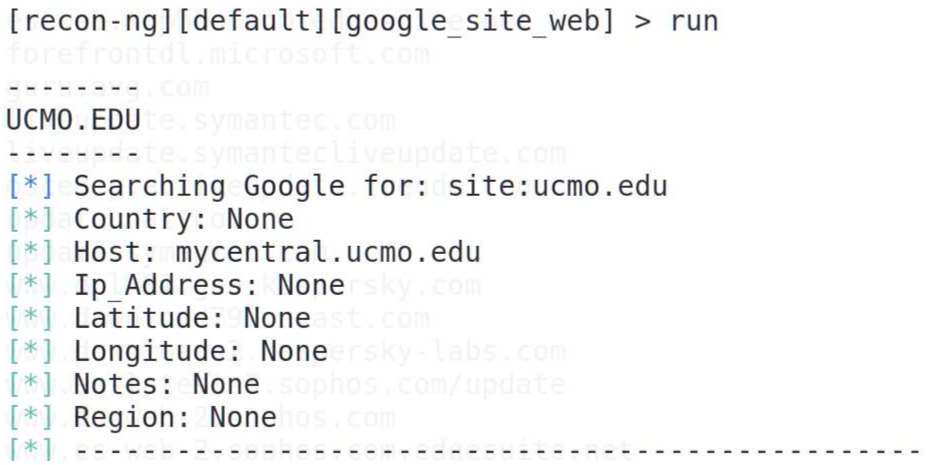
**[recon-ng][UCM]> show domains**

14. Now we are ready the perform the hosts search using the **recon/domains-hosts/google_site_web** module. Type

**[recon-ng][UCM]> modules load recon/domains-hosts/google_site_web**

With all our settings now in place, we can run the module

**[recon-ng] > run**



```
[recon-ng][default][google_site_web] > run

--------
UCMO.EDU
--------
[*] Searching Google for: site:ucmo.edu
[*] Country: None
[*] Host: mycentral.ucmo.edu
[*] Ip_Address: None
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
[*] -------------------------------------------------
```

To review the identified hosts, type

**[recon-ng] > show hosts**

15. Next let’s brute force the domain using the **recon/domains-hosts/brute_hosts** module. Type

**[recon-ng][UCM]> modules load recon/domains-hosts/brute_hosts**

Let's get some info on this module by typing

**[recon-ng] > info**

There is a SOURCE option we need to set. Again, we can brute force the ucmo.edu domain

**[recon-ng] > options set SOURCE ucmo.edu**

We can leave the WORDLIST option to its default value. With all our settings now in place, we can run the module

**[recon-ng] > run**

Obviously, we find quite a few new sub-domains by using this brute force module.

16. To finish the lab, you can exit the Recon-ng tool

```
[recon-ng] > exit
```

You should also clean up the Recon-ng configuration file and database, which are automatically created in your home directory (~)

```
# cd ~
```

```
# rm -rf  .recon-ng
```

This will remove all the information in the database as well as the custom name server configuration you set for Recon-ng.

## Lab Report

- please include your name and 700# at the beginning of your report
- please upload your report to the Blackboard by the due date
- You need to submit a detailed lab report, with screenshots, to describe what you have done and what you have observed
- only word or pdf format is acceptable
- you must show all the necessary commands associated with each task in order to receive credits
- your screenshots size must be appropriate to provide the visible details

In this report, you need to provide the screenshots for the following five tasks.

1. Run the **cache_snoop** module against Microsoft's DNS server. What antivirus software does Microsoft use?
2. Run the **interesting_files** module against a domain of your choice. Provide a screenshot of the module output.
3. Continue from step 14. After obtaining the hostnames from the ucmo.edu domain, we'd like to find the corresponding IP addresses for those identified hosts. We will use the **recon/hosts-hosts/resolve** module. Provide a screenshot of the module output.
4. Run the **recon/profiles-profiles/profiler** module. Set the SOURCE to be your name. Provide a screenshot of the module output.
5. Run the **recon/domains-hosts/bing_domain_web** to against a domain of your choice. Provide a screenshot of the module output.