Open Source Web Reconnaissance with Recon-ng

01 August 2016 • 16 mins read

• Information gathering • OSINT • Recon-ng

During a penetration test, a big part of the success in the exploitation phase depends from how good the information gathering was performed. Since this activity, especially when dealing with a huge amount of informations, is time consuming, it is a good idea to rely on tools which make reconnaissance in automated way.

Recon-ng is an incredibly powerful tool for Open Source Intelligence Gathering (OSINT); actually, it is a reconnaissance framework written in Python built with a Metasploit like usage model (we will see what Metasploit is further on, for now it is enough to know that it is the most famous penetration testing framework).
Reconnaissance is considered as the activity of acquiring open source informations, i.e. available on the Internet, about a target in a passive way (passive reconnaissance); conversely, discovery is the activity which permits to acquire informations by sending packets directly to the target (active reconnaissance). Even if Recon-ng is mainly a passive reconnaissance framework, it includes also some elements for discovery and exploitation.

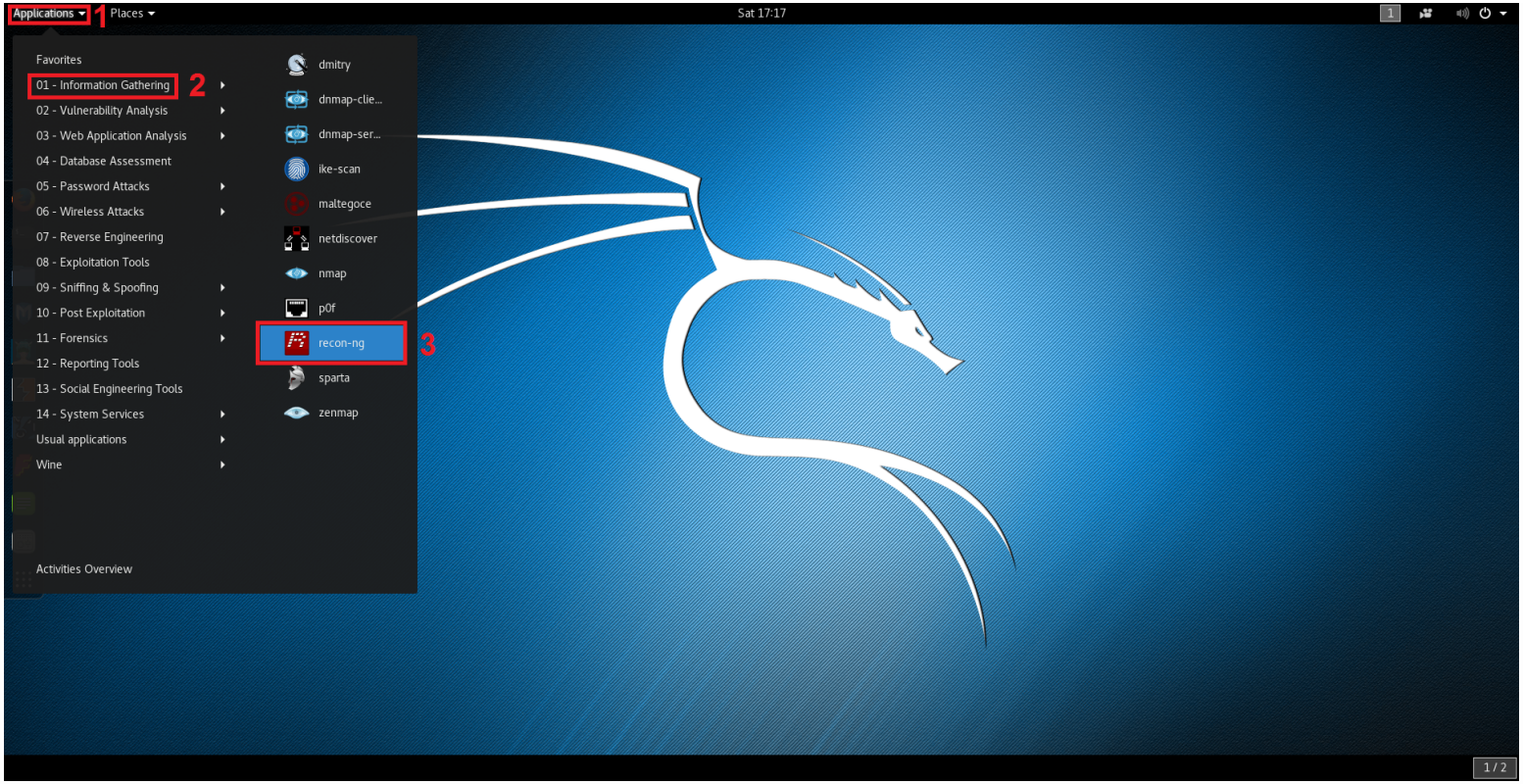
Installation

Since we will use a lot of tools during the next posts, I highly suggest to set up a Virtual Machine with a Penetration Testing distribution installed on. Personally I use VMware Workstation 12 Player as hypervisor for server and desktop virtualization which is free and can be downloaded from the official website. Regarding operating systems, I use mainly Kali Linux, which is a Debian based distribution. This distro is very useful because it has a pretty good number of tools preinstalled and preconfigured leaving to the user a ready to use PT machine. I will not explain how to set up a VM since you can find a lot of tutorials about that on the web.

Anyway, you can still download Recon-ng on your favorite Linux distribution from author repository using `git clone` and installing required dependencies (this is also an option in Kali Linux in case you want the latest version available): https://bitbucket.org/LaNMaSteR53/recon-ng.

Usage

In Kali Linux, we can start Recon-ng in different ways. One is by navigating in the applications menu by clicking on *Applications* > *Information Gathering* > *recon-ng* like shown in the following image:



Same thing can be done by clicking on the “Show application” menu:

Another possibility is launching it by simply opening the Terminal and typing `recon-ng`. In any case, we are prompted with the framework banner, version and number of modules for each category:

Modules are the core of the framework and in the current version there are five categories:

- Recon modules - for reconnaissance activities;
- Reporting modules - for reporting results on a file;
- Import modules - for importing values from a file into a database table;
- Exploitation modules - for explotation activities;
- Discovery modules - for discovery activities.

The good thing is that everyone can implement his own module written in Python and integrate it inside the framework.

Since we are dealing with information gathering, we will focus on recon modules.
The framework accepts commands via command line; to have a list of the commands just type `help` and press enter:

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

Commands (type [help|?] <topic>):
--------------------------------
add             Adds records to the database
back            Exits the current context
delete          Deletes records from the database
exit            Exits the framework
help            Displays this menu
keys            Manages framework API keys
load            Loads specified module
pdb             Starts a Python Debugger session
query           Queries the database
record          Records commands to a resource file
reload          Reloads all modules
resource        Executes commands from a resource file
search          Searches available modules
set             Sets module options
shell           Executes shell commands
show            Shows various framework items
snapshots       Manages workspace snapshots
spool           Spools output to a file
unset           Unsets module options
use             Loads specified module
workspaces      Manages workspaces
```

To display a list of all available modules for each category we can use the `show` command:

```
show modules
```

Since right now we are only interested in recon modules, we can limit the search to them:

```
[recon-ng][default] > show modules recon

  Recon
  -----
    recon/companies-contacts/bing_linkedin_cache
    recon/companies-contacts/indeed
    recon/companies-contacts/jigsaw/point_usage
    recon/companies-contacts/jigsaw/purchase_contact
    recon/companies-contacts/jigsaw/search_contacts
    recon/companies-contacts/linkedin_auth
    recon/companies-multi/github_miner
    recon/companies-multi/whois_miner
    recon/contacts-contacts/mailtester
    recon/contacts-contacts/mangle
    recon/contacts-contacts/unmangle
    recon/contacts-credentials/hibp_breach
    recon/contacts-credentials/hibp_paste
    recon/contacts-domains/migrate_contacts
    recon/contacts-profiles/fullcontact
    recon/credentials-credentials/adobe
    recon/credentials-credentials/bozocrack
    recon/credentials-credentials/hashes_org
    recon/domains-contacts/metacrawler
    recon/domains-contacts/pgp_search
```

```
    recon/domains-contacts/whois_pocs
    recon/domains-credentials/pwnedlist/account_creds
    recon/domains-credentials/pwnedlist/api_usage
    recon/domains-credentials/pwnedlist/domain_creds
    recon/domains-credentials/pwnedlist/domain_ispwned
    recon/domains-credentials/pwnedlist/leak_lookup
    recon/domains-credentials/pwnedlist/leaks_dump
    recon/domains-domains/brute_suffix
    recon/domains-hosts/bing_domain_api
    recon/domains-hosts/bing_domain_web
    recon/domains-hosts/brute_hosts
    recon/domains-hosts/builtwith
    recon/domains-hosts/google_site_api
    recon/domains-hosts/google_site_web
    ..................................
```

The structure for each module is the following:

```
module-category/data-conversion/module-name
```

Consider, for example, `recon/domains-hosts/google_site_web`: this performs a recon activity using Google Search Engine to convert an information about a domain into data about hosts of that domain. Keep in mind that certain modules require valid API key to run; some keys can be acquired by simply registering on the related website.
To select a module we need the `use` command:

```
use recon/domains-hosts/google_site_web
```

Once the module is selected we can show informations about it:

```
[recon-ng][default][google_site_web] > show info

      Name: Google Hostname Enumerator
      Path: modules/recon/domains-hosts/google_site_web.py
    Author: Tim Tomes (@LaNMaSteR53)

Description:
  Harvests hosts from Google.com by using the 'site' search operator. Updates the 'h
  the results.

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

Source Options:
  default        SELECT DISTINCT domain FROM domains WHERE domain 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
```

In this way we can read the description and take a look at the options we can set before running the recon activity. As you can see, the action performed by this module is pretty the same as the one explained in the article Information gathering with Google Search Engine, but this time it is done in an automated way.
In case we want to analyze module source code we can either use `show source` or navigate to `/usr/share/recon-ng/modules/recon/domains-hosts` where the python file `google_site_web.py` is

located (note that folders structure reflects modules categories and data conversions).
Once all required options are set up through `set` command, the module can be executed with `run`.

We will see now an example of reconnaissance activity performed on the National Institute of Standards and Technology (NIST) domain.
Before starting, we need to introduce the concept of workspace: Recon-ng allows to define a workspace for each target subject of reconnaissance; by doing this, it will create a database containing all gathered informations about the target itself. This is the reason why in the “framework help” shown before there is the `query` command, which allows to examine the DB using Standard Query Language (SQL), and also why import modules are present.

We start by creating a new workspace:

```
workspaces add NIST
```

After that, the command line shows the change from the default workspace to the new one. Then we need to associate a domain with the created workspace and finally we can check that everything is set up correctly by listing domains with `show`:

```
[recon-ng][default] > workspaces add NIST
[recon-ng][NIST] > add domains nist.gov
[recon-ng][NIST] > show domains

  +---------------------------------+
  | rowid |  domain  |    module    |
  +---------------------------------+
  | 1     | nist.gov | user_defined |
  +---------------------------------+

[*] 1 rows returned
```

Same result can be obtained with:

```
[recon-ng][NIST] > query select * from domains
```

This can be checked also by querying the database with an external tool; the DB is located in the following folder:

```
~/.recon-ng/workspaces/NIST
```

Here there is a file called `data.db` which is the database for NIST workspace; to explore the DB we can use the tool `sqlite3` already installed in Kali Linux:

```
root@kali:~/.recon-ng/workspaces/NIST# sqlite3 data.db
SQLite version 3.13.0 2016-05-18 10:57:30
Enter ".help" for usage hints.
sqlite> select * from domains;
nist.gov|user_defined
```

To exit from the program, just type `.exit`.

We can also add a company name:

```
[recon-ng][NIST] > add companies
company (TEXT): NIST
description (TEXT): National Institute of Standards and Technology
[recon-ng][NIST] > show companies

  +-------------------------------------------------------------------------------
```

```
  | rowid | company |                  description                  |    module    |
  +-------+---------+-----------------------------------------------+--------------+
  | 1     | NIST    | National Institute of Standards and Technology | user_defined |
  +-------+---------+-----------------------------------------------+--------------+

[*] 1 rows returned
```

Adding domains and companies is the initial step because they are inputs used by modules to perform information gathering. To check all modules using these two informations as a starting point we can leverage the `search` command:

```
[recon-ng][NIST] > search domains-
[*] Searching for 'domains-'...

  Recon
  -----
    recon/domains-contacts/metacrawler
    recon/domains-contacts/pgp_search
    recon/domains-contacts/whois_pocs
    recon/domains-credentials/pwnedlist/account_creds
    recon/domains-credentials/pwnedlist/api_usage
    recon/domains-credentials/pwnedlist/domain_creds
    recon/domains-credentials/pwnedlist/domain_ispwned
    recon/domains-credentials/pwnedlist/leak_lookup
    recon/domains-credentials/pwnedlist/leaks_dump
    recon/domains-domains/brute_suffix
    recon/domains-hosts/bing_domain_api
    recon/domains-hosts/bing_domain_web
    recon/domains-hosts/brute_hosts
```

```
    recon/domains-hosts/builtwith
    recon/domains-hosts/google_site_api
    recon/domains-hosts/google_site_web
    recon/domains-hosts/hackertarget
    recon/domains-hosts/netcraft
    recon/domains-hosts/shodan_hostname
    recon/domains-hosts/ssl_san
    recon/domains-hosts/vpnhunter
    recon/domains-vulnerabilities/ghdb
    recon/domains-vulnerabilities/punkspider
    recon/domains-vulnerabilities/xssed
    recon/domains-vulnerabilities/xssposed

[recon-ng][NIST] > search companies-
[*] Searching for 'companies-'...

  Recon
  -----
    recon/companies-contacts/bing_linkedin_cache
    recon/companies-contacts/indeed
    recon/companies-contacts/jigsaw/point_usage
    recon/companies-contacts/jigsaw/purchase_contact
    recon/companies-contacts/jigsaw/search_contacts
    recon/companies-contacts/linkedin_auth
    recon/companies-multi/github_miner
    recon/companies-multi/whois_miner
```

Suppose we want to start populating our DB with hostnames related to *nist.gov* domain usign `google_site_web` module seen before; to check parameters required to run it we can display module

options:

```
[recon-ng][NIST][google_site_web] > show options

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

Since we have already set the domain, the “Current Value” which says “default” is taken directly from the DB. Then, we can just run the module and after a little while we get the results:

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

--------
NIST.GOV
--------
[*] Searching Google for: site:nist.gov
[*] [host] www.nsrl.nist.gov (<blank>)
[*] [host] gams.nist.gov (<blank>)
[*] [host] physics.nist.gov (<blank>)
[*] [host] face.nist.gov (<blank>)
[*] [host] scap.nist.gov (<blank>)
[*] [host] patapsco.nist.gov (<blank>)
[*] [host] nvd.nist.gov (<blank>)
[*] [host] kinetics.nist.gov (<blank>)
[*] [host] srdata.nist.gov (<blank>)
[*] [host] www.cftt.nist.gov (<blank>)
[*] [host] cccbdb.nist.gov (<blank>)
[*] [host] museum.nist.gov (<blank>)
[*] [host] thermosymposium.nist.gov (<blank>)
```

```
[*] [host] www.atp.nist.gov (<blank>)
[*] [host] www.ctcms.nist.gov (<blank>)
[*] [host] usgcb.nist.gov (<blank>)
[*] [host] www.nist.gov (<blank>)
[*] [host] trecvid.nist.gov (<blank>)
[*] [host] stonewall.nist.gov (<blank>)
.....................................
-------
SUMMARY
-------
[*] 73 total (73 new) hosts found.
```

In this case we discovered 73 hosts related to the domain; we can show the list of discovered host:

```
[recon-ng][NIST][google_site_web] > show hosts

  +-------------------------------------------------------------------------------
  | rowid |             host             | ip_address | region | country | latitud
  +-------------------------------------------------------------------------------
  | 1     | www.nsrl.nist.gov            |            |        |         |
  | 2     | gams.nist.gov                |            |        |         |
  | 3     | physics.nist.gov             |            |        |         |
  | 4     | face.nist.gov                |            |        |         |
  | 5     | scap.nist.gov                |            |        |         |
  | 6     | patapsco.nist.gov            |            |        |         |
  | 7     | nvd.nist.gov                 |            |        |         |
  | 8     | kinetics.nist.gov            |            |        |         |
  | 9     | srdata.nist.gov              |            |        |         |
  | 10    | www.cftt.nist.gov            |            |        |         |
  | 11    | cccbdb.nist.gov              |            |        |         |
```

```
  | 12    | museum.nist.gov               |           |        |         |
  | 13    | thermosymposium.nist.gov      |           |        |         |
  | 14    | www.atp.nist.gov              |           |        |         |
  | 15    | www.ctcms.nist.gov            |           |        |         |
  | 16    | usgcb.nist.gov                |           |        |         |
  | 17    | www.nist.gov                  |           |        |         |
  | 18    | trecvid.nist.gov              |           |        |         |
  | 19    | stonewall.nist.gov            |           |        |         |
.....................................................................................
```

As the table shows, we have empty columns ready to store additional informations for each host: these can be populated by hand or by running other modules using host informations we just gathered:

```
[recon-ng][NIST][google_site_web] > search hosts-
[*] Searching for 'hosts-'...

  Recon
  -----
    recon/hosts-domains/migrate_hosts
    recon/hosts-hosts/bing_ip
    recon/hosts-hosts/freegeoip
    recon/hosts-hosts/ipinfodb
    recon/hosts-hosts/resolve
    recon/hosts-hosts/reverse_resolve
    recon/hosts-hosts/ssltools
    recon/hosts-locations/migrate_hosts
    recon/hosts-ports/shodan_ip
```

We can find IP addresses for each host by running `recon/hosts-hosts/resolve` module, while the geolocation can be acquired with `recon/hosts-hosts/freegeoip`:

```
[recon-ng][NIST][freegeoip] > show hosts

  +-------------------------------------------------------------------------------
  | rowid |             host              |   ip_address   |         region
  +-------------------------------------------------------------------------------
  | 1     | www.nsrl.nist.gov             | 129.6.24.57    | Gaithersburg, Maryland
  | 2     | gams.nist.gov                 | 129.6.24.27    | Gaithersburg, Maryland
  | 3     | physics.nist.gov              | 129.6.13.152   | Gaithersburg, Maryland
  | 4     | face.nist.gov                 | 132.163.4.217  | Boulder, Colorado
  | 5     | scap.nist.gov                 | 129.6.13.177   | Gaithersburg, Maryland
  | 6     | patapsco.nist.gov             | 129.6.13.93    | Gaithersburg, Maryland
  | 7     | nvd.nist.gov                  | 129.6.13.177   | Gaithersburg, Maryland
  | 8     | kinetics.nist.gov             | 129.6.24.48    | Gaithersburg, Maryland
  | 9     | srdata.nist.gov               | 129.6.13.111   | Gaithersburg, Maryland
  | 10    | www.cftt.nist.gov             | 129.6.24.57    | Gaithersburg, Maryland
  | 11    | cccbdb.nist.gov               | 129.6.13.59    | Gaithersburg, Maryland
  | 12    | museum.nist.gov               | 129.6.13.111   | Gaithersburg, Maryland
  | 13    | thermosymposium.nist.gov      | 132.163.4.124  | Boulder, Colorado
  | 14    | www.atp.nist.gov              | 132.163.4.217  | Boulder, Colorado
  | 15    | www.ctcms.nist.gov            | 129.6.24.51    | Gaithersburg, Maryland
  | 16    | usgcb.nist.gov                | 129.6.13.177   | Gaithersburg, Maryland
  | 17    | www.nist.gov                  | 132.163.4.18   | Boulder, Colorado
  | 18    | trecvid.nist.gov              | 132.163.4.217  | Boulder, Colorado
  | 19    | stonewall.nist.gov            | 129.6.13.93    | Gaithersburg, Maryland
.................................................................................
```

As shown, in minutes we have acquired tons of informations about target hosts.
Now we can lower the search level by digging even deeper: what about looking for contact informations such as names and email addresses? We can achieve this objective by running `recon/domains-contacts/pgp_search`: in fact as the description reports, this module searches the MIT public PGP key server for email addresses of the given domain. After module has been executed, we can display results stored in the DB (of course names and addresses in the following table are fictional for privacy reasons):

```
[recon-ng][NIST][pgp_search] > show contacts

  +-------------------------------------------------------------------------------
  | rowid |   first_name  | middle_name |    last_name     |            email
  +-------------------------------------------------------------------------------
  | 1     | Bugs          |             | Bunny            | bugs.bunny@nist.gov
  | 2     | Foghorn       |             | Leghorn          | foghorn.leghorn@nist.go
  | 3     | Daffy         |             | Duck             | daffy.duck@nist.gov
```

This is not over yet: we can also search if those contacts have been involved in a databreach, like Adobe one in 2013. For this purpose there are two interesting modules, `recon/contacts-credentials/hibp_breach` and `recon/contacts-credentials/hibp_paste`: the first one leverages haveibeenpwned.com API to determine if email addresses are associated with breached credentials, while the other one uses the API to determine if email addresses have been published to various paste sites.

You can check if your email address has been compromised in data breaches by simply going on the *Have I Been Pwned?* (HIBP) website and launching a search. This service collects and analyzes database dumps and pastes leaked by data breaches happened over the years regarding millions of accounts.

All these informations can be useful during next phases of the attack, especially for Social Engineering (we will look into this technique in future articles).

Once collected enough informations, it is useful to report them in a document. Fortunately, Recon-ng offers modules to report results in different formats:

```
[recon-ng][NIST] > show modules reporting

  Reporting
  ---------
    reporting/csv
    reporting/html
    reporting/json
    reporting/list
    reporting/pushpin
    reporting/xlsx
    reporting/xml
```

For example, we can choose to save the returns in an HTML page file:

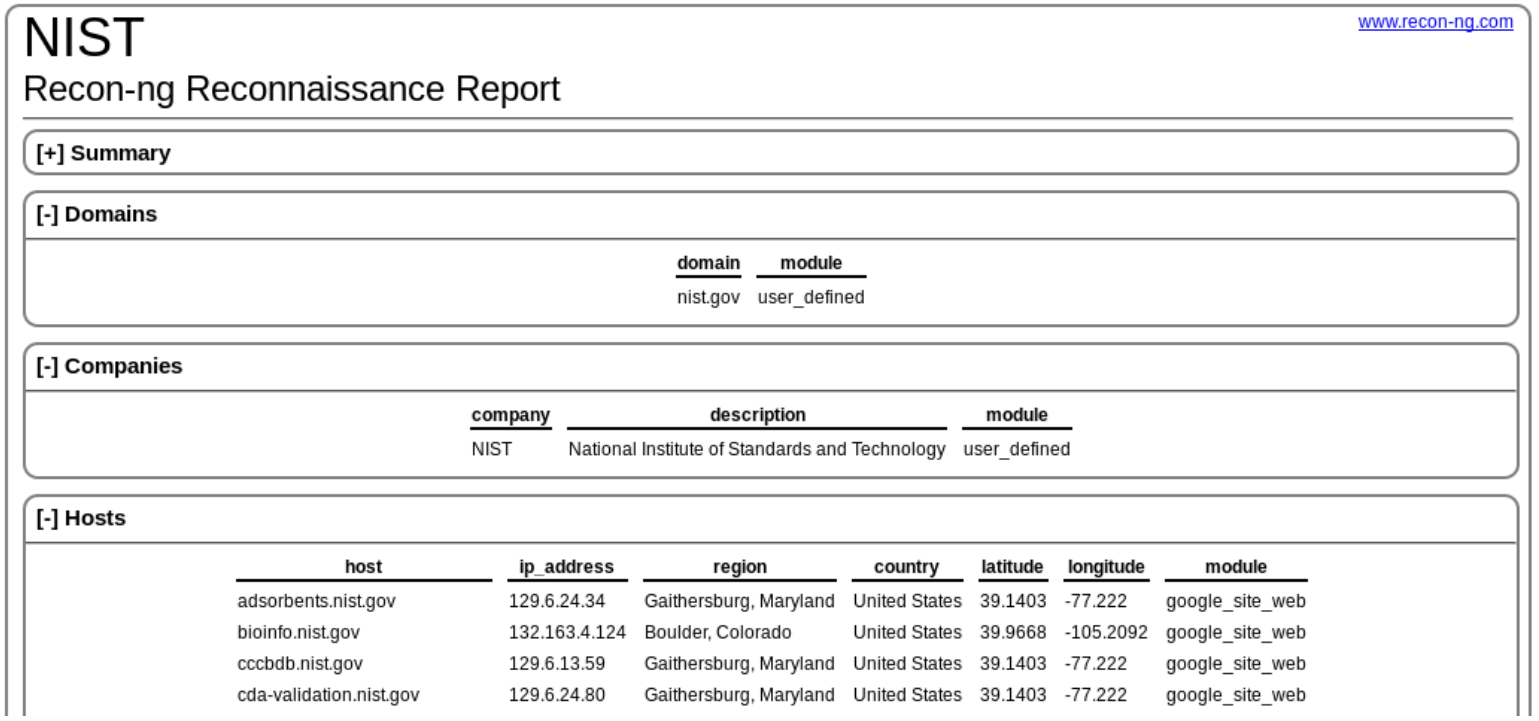
```
[recon-ng][NIST] > use reporting/html
[recon-ng][NIST][html] > show options

  Name      Current Value                                  Required  Description
  --------  -------------                                  --------  -----------
  CREATOR                                                  yes       creator name for
  CUSTOMER                                                 yes       customer name fo
  FILENAME  /root/.recon-ng/workspaces/NIST/results.html   yes       path and filenam
  SANITIZE  True                                           yes       mask sensitive d

[recon-ng][NIST][html] > set CREATOR Spread Security
CREATOR => Spread Security
[recon-ng][NIST][html] > set CUSTOMER NIST
```

```
CUSTOMER => NIST
[recon-ng][NIST][html] > run
[*] Report generated at '/root/.recon-ng/workspaces/NIST/results.html'.
```

Results can be then visualized using a common web browser:



NIST

www.recon-ng.com

Recon-ng Reconnaissance Report

[+] Summary

[-] Domains

domain	module
nist.gov	user_defined

[-] Companies

company	description	module
NIST	National Institute of Standards and Technology	user_defined

[-] Hosts

host	ip_address	region	country	latitude	longitude	module
adsorbents.nist.gov	129.6.24.34	Gaithersburg, Maryland	United States	39.1403	-77.222	google_site_web
bioinfo.nist.gov	132.163.4.124	Boulder, Colorado	United States	39.9668	-105.2092	google_site_web
cccbdb.nist.gov	129.6.13.59	Gaithersburg, Maryland	United States	39.1403	-77.222	google_site_web
cda-validation.nist.gov	129.6.24.80	Gaithersburg, Maryland	United States	39.1403	-77.222	google_site_web

Conclusions

Recon-ng is a valuable framework for reconnaissance which has a really good system for storing and managing data for later use.
We have seen only a small part of its real capabilities, so take your time to explore and experiment with it to take advantage of its true power.