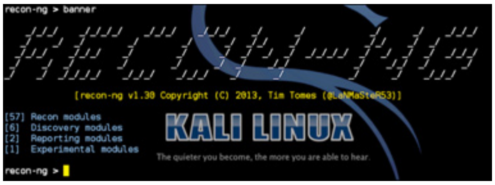


```
recon-ng > banner

        [recon-ng v1.30 Copyright (C) 2013, Tim Tomes (@LaNMaSteR53)]

[57] Recon modules
[6]  Discovery modules
[2]  Reporting modules
[1]  Experimental modules

recon-ng >
```

Recon-ng

By Russ McRee – ISSA Senior Member, Puget Sound (Seattle), USA Chapter

Prerequisites/dependencies

Python interpreter-enabled system, Kali Linux utilized for this review.

The community of tools and developers converges again this month as we explore Tim Tomes' Recon-ng. Jeremy Druin, whose NOWASP Mutillidae we explored in August 2012's *toolsmith*,[1] introduced me to Tim, having recognized another great tool worthy of exploration and sharing with *toolsmith* nation. Recon-ng is optimized for use during the reconnaissance phase of web application penetration testing. You'll note convergence again, given that we described managing web application penetration testing phases in last month's *toolsmith* regarding Redmine. Tim says it best on his Recon-ng site[2]: "Recon-ng is not intended to compete with existing frameworks, as it is designed exclusively for web-based open source reconnaissance. If you want to exploit, use the Metasploit framework. If you want to social engineer, us the Social Engineer Toolkit.[3] If you want to conduct reconnaissance, use Recon-ng!"

More from Tim on Recon-ng, shared exclusively with *toolsmith*:

"Recon-ng is commonly seen as being most useful in the role of supporting social engineering engagements, but the real power of the framework lies in its ability to perform all steps of the traditional penetration testing methodology, except exploitation, within the context of reconnaissance. What does that mean? It means that we can do scope validation through host discovery, server enumeration, vulnerability discovery, and gain access to authentication credentials, all without sending a single packet to the target application or network. Recon-ng does this by leveraging powerful, third-party, web-based resources that do all of this stuff for us and provide access to the results. It is important to keep in mind that there are caveats to this. Using third parties to collect data on clients may be in direct violation of non-disclosure agreements (NDA) or contracts. It is up to the tester to make sure that the client specifically approves this activity as part of the testing agreement.

While the framework is named for its focus on reconnaissance, the intent is not to limit its functionality to only recon. Python developers have been waiting a long time for a fun, easy, and useful project to contribute to. They now have that in Recon-ng. Therefore, when contributors come up with new ideas for modules that cross the boundary of reconnaissance into active discovery and exploitation, they are encouraged to submit them for review. The several discovery modules included in the framework are good examples of this.

I get asked quite often, How does Recon-ng fit into your testing methodology? The answer is simple. It's the first tool I use on every engagement, and often during the scoping process. Do I run every module in the framework? No. It largely depends on the type of assessment. But there are several things I always do. I always harvest hosts from Google, Shodan, and IP-Neighbors and enumerate with the Resolve, BuiltWith, and PunkSPIDER modules. I always harvest contacts using Jigsaw, LinkedIn, PGP, and Whois and mangle them into email addresses with the Mangle module. And I always check for compromised accounts and harvest any available credentials using the various PwnedList modules."

We've provided much detail on the web application penetration testing methodology as describe by SANS in earlier *toolsmiths*, so in order to broaden our horizons a bit, I'll plug Recon-ng use into the various phases of the OWASP Testing Guide v4[4]. Version 4 is the draft version; version 3 (2008) is considered stable. The information gathering section of the guide is a ten-part contribution to section 4 of the guide, *Web Application Penetration Testing*. Immediately relevant steps from the draft TOC include:

- 4.2.1 Testing for Web Server Fingerprint (OWASP-IG-004)
- 4.2.2 Review Webserver Metafiles (OWASP-IG-001)
- 4.2.5 Identify application entry points (OWASP-IG-003)

We'll also use reconnaissance methods to lend to section 4.4.2 Testing for User Enumeration and Guessable User Account (OWASP-AT-002) from 4.4 Authentication Testing.

We'll use Recon-ng to realize the goals of a few of these OWASP Testing Guide steps as we explore further below.

Recon-ng installation

Recon-ng installs with ease on any Python- and Git-enabled system. On Kali, running as root; it's as simple as:

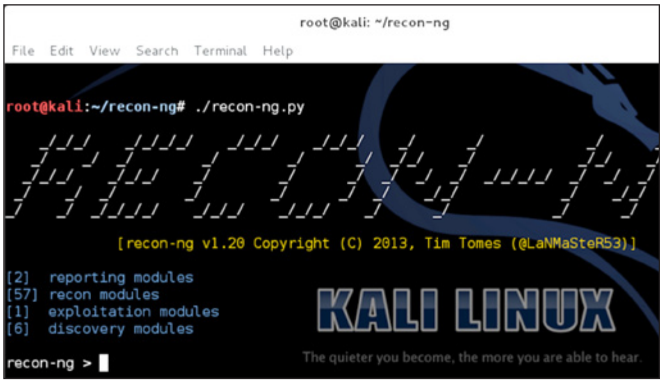
```
git clone https://LaNMaSteR53@bitbucket.org/
LaNMaSteR53/recon-ng.git
```

[1] http://holisticinfosec.org/toolsmith/pdf/august2012.pdf.

[2] http://www.recon-ng.com.

[3] http://holisticinfosec.org/toolsmith/pdf/february2013.pdf.

[4] https://www.owasp.org/index.php/OWASP_Testing_Guide_v4_Table_of_Contents.


root@kali: ~/recon-ng
File Edit View Search Terminal Help
KALI LINUX
The quieter you become, the more you are able to hear.

```
root@kali:~/recon-ng# ./recon-ng.py

            [recon-ng v1.20 Copyright (C) 2013, Tim Tomes (@LaNMaSteR53)]

[2]  reporting modules
[57] recon modules
[1]  exploitation modules
[6]  discovery modules

recon-ng >
```

Figure 1 – Getting underway with Recon-ng

```
cd recon-ng
./recon-ng.py
```

Figure 1 represents the initiated Recon-ng shell and its 57 recon, six discovery, one exploitation, and two reporting modules.

The dependencies on dnspython, httplib2, and python-oauth2 are already met in the recon-ng lib directory. If you're familiar with Metasploit, you'll be right at home with Recon-ng. Refer to the wiki[5] for a usage overview. I worked with both the stable version 1.20, as well as the beta of 1.30, which should be a stable release by the time you read this. 1.30 includes major updates including what @LaNMaSteR53 tweeted is a badly needed API key handling system.

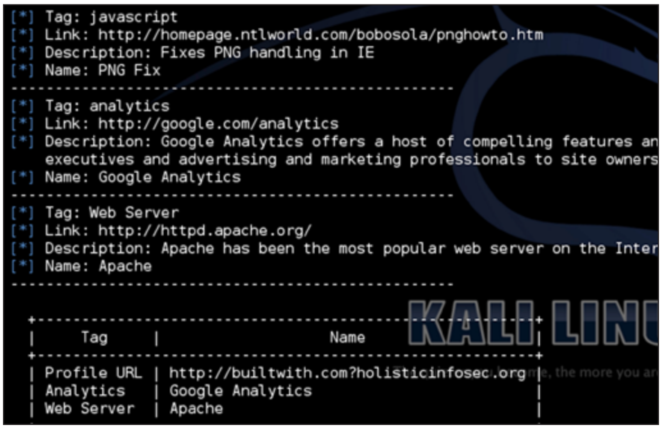
Putting Recon-ng to use

A few quick use pointers may help you get under way with Recon-ng. Command completion is handy as you consider typing commands such as

```
use recon/contacts/enum/http/should_change_
password.
```

Hitting *tab* while keying will complete, based on options for command or parameter. Also extraordinarily useful is the *smart load* feature, which loads modules when you refer to a keyword unique to the desired module's name. As an example, for the first module we'll test I simply typed *use xssed* which loaded the *recon/hosts/enum/http/web/xssed* module. This works well without the full path as it is the only module containing the string *xssed,* but if multiple modules share the same keyword, you'll receive a list of possible modules. *Use* is also, in reality, an alias for the *load* command; they work identically. Apparently, overly sensitive Metasploit users bugged Tim until he created command alignment. From a Recon-ng prompt the best way to see all modules available to you is to pass the show modules command, and don't forget to use the ? command when you need more information. As an example, *show ?* reveals your usage options are show *[modules|options|workspaces|schema|<table>].* With a particular module loaded, use *info* for name, author, description, and options details. Then use *set* based on the options defined followed by the *run* command. That's all there is to it. You can define individual workspaces or other global options as well. I ran *show options*, then *set workspace holisticinfosec* for our efforts here. You can also set proxy settings here if you wish to record your sessions with the like of Burp Suite. The resulting report from Burp is a nice output product for your pentest engagements. Equally useful might be the use of an anonymizing proxy.

Use the recon/hosts/enum/http/api/builtwith module for 4.2.1 Testing for Web Server Fingerprint (OWASP-IG-004). As the guidance states "knowing the version and type of a running web server allows testers to determine known vulnerabilities and the appropriate exploits to use during testing" – you can imagine why. I loaded the module, passed *set host holisticinfosec.org*, and followed with *run*, resulting in Figure 2.



```
[*] Tag: javascript
[*] Link: http://homepage.ntlworld.com/bobosola/pnghowto.htm
[*] Description: Fixes PNG handling in IE
[*] Name: PNG Fix
--------------------------------------------------
[*] Tag: analytics
[*] Link: http://google.com/analytics
[*] Description: Google Analytics offers a host of compelling features an
    executives and advertising and marketing professionals to site owners
[*] Name: Google Analytics
--------------------------------------------------
[*] Tag: Web Server
[*] Link: http://httpd.apache.org/
[*] Description: Apache has been the most popular web server on the Inter
[*] Name: Apache
--------------------------------------------------

  +-----------------------------------------------------+
  |    Tag      |                Name                   |
  +-----------------------------------------------------+
  | Profile URL | http://builtwith.com?holisticinfosec.org |
  | Analytics   | Google Analytics                      |
  | Web Server  | Apache                                |
```

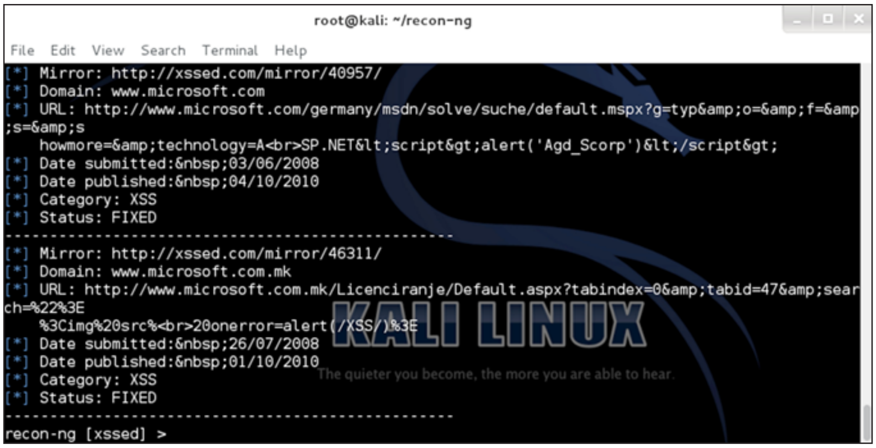
Figure 2 – Recon-ng establishes server details

For 4.2.2 Review Webserver Metafiles (OWASP-IG-001),[6] an ideal module is discovery/info_disclosure/http/interesting_files. This is not a passive module; it will reach out and touch the defined source, and download discovered files such as robots.txt, sitemap.xml, crossdomain.xml, and phpinfo.php. The discovered and downloaded files are written to the workspace directory in which you are operating. The */recon-ng/workspace/default* workspace is the default if none is specified in the global options.

The xssed module relates nicely to section 4.2.5 Identify application entry points (OWASP-IG-003), which describes the process to identify application entry points. OWASP's brief overview of this phase states that "enumerating the application and its attack surface is a key precursor before any attack should commence. This section will help you identify and map out every area within the application that should be investigated once your enumeration and mapping phase has been completed." Parameters vulnerable to cross-site scripting (XSS) via GET or POST requests certainly fall in the "worthy of investigation" category as variables exhibiting XSS vulnerabilities are sometime vulnerable to other issues such as SQL injection or directory traversal. Of course, XSS

[5] http://www.recon-ng.com/wiki/Home - !getting-started.

[6] https://www.owasp.org/index.php/Testing:_Spiders,_Robots,_and_Crawlers_%28OWASP-IG-001%29.



```
root@kali: ~/recon-ng
File  Edit  View  Search  Terminal  Help
[*] Mirror: http://xssed.com/mirror/40957/
[*] Domain: www.microsoft.com
[*] URL: http://www.microsoft.com/germany/msdn/solve/suche/default.mspx?g=typ&amp;o=&amp;f=&amp
;s=&amp;s
    howmore=&amp;technology=A<br>SP.NET&lt;script&gt;alert('Agd_Scorp')&lt;/script&gt;
[*] Date submitted:&nbsp;03/06/2008
[*] Date published:&nbsp;04/10/2010
[*] Category: XSS
[*] Status: FIXED
--------------------------------------------------
[*] Mirror: http://xssed.com/mirror/46311/
[*] Domain: www.microsoft.com.mk
[*] URL: http://www.microsoft.com.mk/Licenciranje/Default.aspx?tabindex=0&amp;tabid=47&amp;sear
ch=%22%3E
    %3Cimg%20src%<br>20onerror=alert(/XSS/)%3E
[*] Date submitted:&nbsp;26/07/2008
[*] Date published:&nbsp;01/10/2010
[*] Category: XSS
[*] Status: FIXED
--------------------------------------------------
recon-ng [xssed] >
```

Figure 3 – Recon-ng XSSed module results

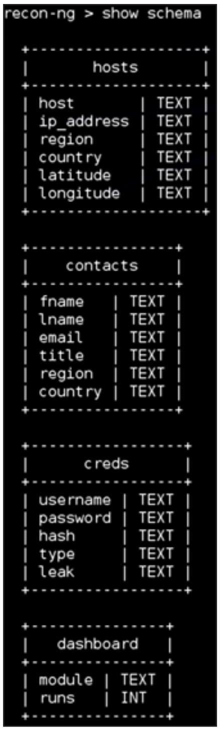
in and of itself represents a number of opportunities for the attacker and should be paid close attention as such.

The xssed module as written by Micah Hoffman (@WebBreacher) checks XSSed.com for XSS records for the given domain and displays the first 20 results. From the Recon-ng prompt I passed the *use xssed* command followed by *set domain microsoft.com*. Given that I work there and my attack and penetration testing may have a Microsoft domain in scope for a penetration test, this module could prove a logical first step. Note that all the returned results for this effort have been fixed, even if results state otherwise. After setting the domain parameter one need only issue a *run* command to kick off the module. Figure 3 shows the results.

The result advises us that, had it not been fixed, the search parameter would have been ideal for further exploration or use in packaging XSS payloads during an exploitation phase.

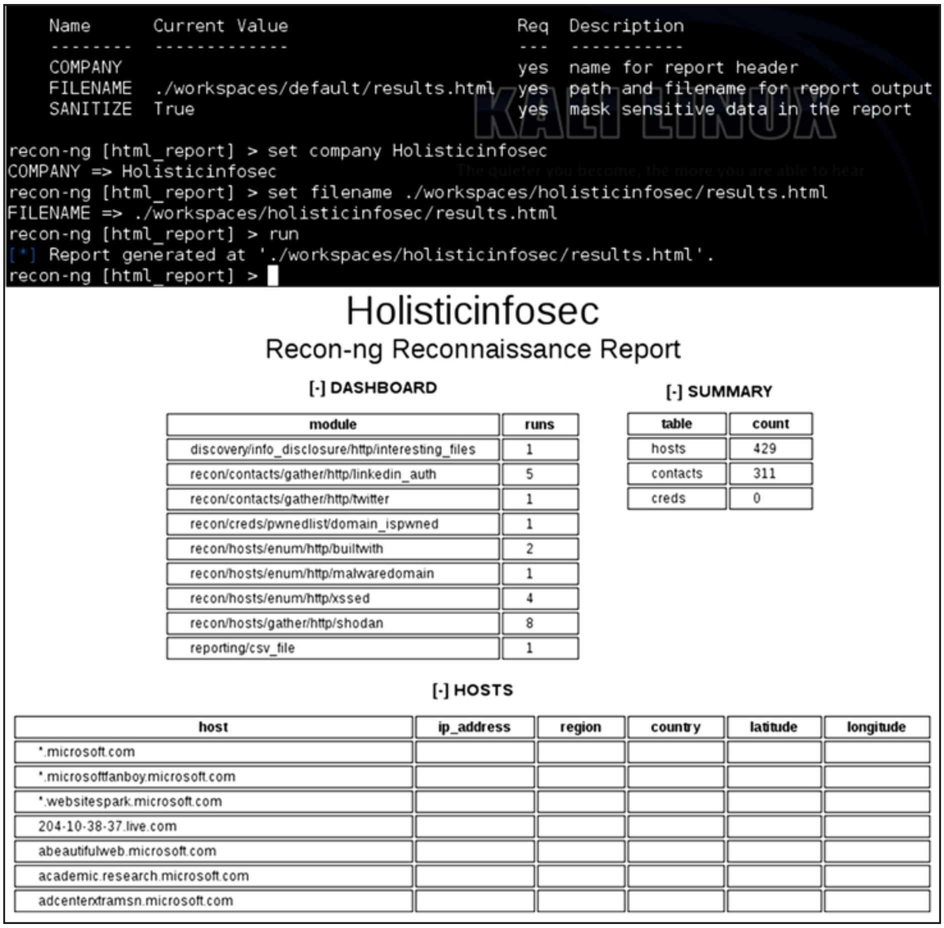
Recon-ng's LinkedIn Authenticated Contact Enumerator is a great way to gather possible social engineering or bruteforcing targets, ideal during the 4.4.2 Testing for User Enumeration and Guessable User Account (OWASP-AT-002) phase. You'll need a LinkedIn API key; just login with you LinkedIn cred and visit the LinkedIn Developer Network.[7] Note: a few Recon-ng modules require API keys. Keep in mind that the Pwnedlist API has a rather high cost associated with it, but if your organization has already purchased API access, you can leverage it with Recon-ng for the Pwnedlist modules *account_creds, api_usage, domain_creds, domain_ispwned, leak_lookup,* and *leaks_dump*. Tim pointed out that, as a Pwnedlist customer, he extremely fond of the *domain_creds* module in particular as it returns actual domain credentials. Nothing like walking in to a customer penetration testing engagement already in possession of domain creds. For the LinkedIn module run *use linkedin*, followed by *set company <target>*, then *run*. No screenshot here as the module dumps lots of juicy contact data and I don't want a bunch of folks upset with me.

Keep in mind that you can always query the native Recon-ng SQLite database with the *query* command followed by common SQL syntax. As an example query *select * from hosts* returns data



```
recon-ng > show schema

+-----------------+
|      hosts      |
+-----------------+
| host      | TEXT |
| ip_address | TEXT |
| region    | TEXT |
| country   | TEXT |
| latitude  | TEXT |
| longitude | TEXT |
+-----------------+

+---------------+
|   contacts    |
+---------------+
| fname   | TEXT |
| lname   | TEXT |
| email   | TEXT |
| title   | TEXT |
| region  | TEXT |
| country | TEXT |
+---------------+

+---------------+
|     creds     |
+---------------+
| username | TEXT |
| password | TEXT |
| hash     | TEXT |
| type     | TEXT |
| leak     | TEXT |
+---------------+

+--------------+
|  dashboard   |
+--------------+
| module | TEXT |
| runs   | INT  |
+--------------+
```

Figure 4 – Recon-ng database schema



```
    Name      Current Value                        Req  Description
    --------  -------------                        ---  -----------
    COMPANY                                        yes  name for report header
    FILENAME  ./workspaces/default/results.html   yes  path and filename for report output
    SANITIZE  True                                 yes  mask sensitive data in the report

recon-ng [html_report] > set company Holisticinfosec
COMPANY => Holisticinfosec
recon-ng [html_report] > set filename ./workspaces/holisticinfosec/results.html
FILENAME => ./workspaces/holisticinfosec/results.html
recon-ng [html_report] > run
[*] Report generated at './workspaces/holisticinfosec/results.html'.
recon-ng [html_report] >
```

Holisticinfosec

Recon-ng Reconnaissance Report

[-] DASHBOARD

module	runs
discovery/info_disclosure/http/interesting_files	1
recon/contacts/gather/http/linkedin_auth	5
recon/contacts/gather/http/twitter	1
recon/creds/pwnedlist/domain_ispwned	1
recon/hosts/enum/http/builtwith	2
recon/hosts/enum/http/malwaredomain	1
recon/hosts/enum/http/xssed	4
recon/hosts/gather/http/shodan	8
reporting/csv_file	1

[-] SUMMARY

table	count
hosts	429
contacts	311
creds	0

[-] HOSTS

host	ip_address	region	country	latitude	longitude
*.microsoft.com					
*.microsoftfanboy.microsoft.com					
*.websitespark.microsoft.com					
204-10-38-37.live.com					
abeautifulweb.microsoft.com					
academic.research.microsoft.com					
adcenterxtramsn.microsoft.com					

Figure 5 – Recon-ng database schema

[7] https://www.linkedin.com/secure/developer.

populated in the columns host, ip_address, region, country, latitude, and longitude during module runs. The database schema is included in Figure 4.

Finally, you will definitely want to take advantage of the reporting modules.

Tim mentioned that the reporting/csv_file module is great for importing into Excel then massaging the data, while reporting/html_report module is optimal for producing reports for customers. Figure 5 shows my reporting run against all data I'd written for the db.

There are, as is often the case with great *toolsmith* topics, too many features and killer use case scenarios to cover here. I even suggested to Tim he write the Recon-ng book. Yes, I think it's that good.

In conclusion

I'm really excited about Recon-ng and wish Tim great success. My two favorite phases are reconnaissance and exploitation, and Recon-ng fits the bill to dominate the first and contribute greatly to the second. Setting it up and getting started is a sixty-second proposition and leaves you no room for excuses. Get cracking with this tool STAT. Run it against entities specific to your organizations and immediately benefit. Or there's always the alternative of waiting and having the hackers do it for you.

Ping me via email if you have questions or suggestions for a topic via russ at holisticinfosec dot org or hit me on Twitter @ holisticinfosec.

Cheers…until next month.

About the Author

Russ McRee manages the Security Analytics team (security incident management, penetration testing, monitoring) for Microsoft's Online Services Security & Compliance organization. In addition to toolsmith, he's written for numerous other publications, speaks regularly at events such as DEFCON, Black Hat, and RSA, and is a SANS Internet Storm Center handler. As an advocate for a holistic approach to the practice of information assurance Russ maintains holisticinfosec.org. He serves in the Washington State Guard as the Cybersecurity Advisor to the Washington Military Department. Reach him at russ at holisticinfosec dot org or @holisticinfosec.