Faraday IPE: When Tinfoil Won't Work for Pentesting

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

Prerequisites

Typically *nix, tested on Debian, Ubuntu, Kali, etc.

Kali 1.1.0 recommended, virtual machine or physical

Overview

I love me some tinfoil-hat-wearing conspiracy theorists, nothing better than sparking up a lively conversation with a "Hey man, what was that helicopter doing over your house?" and you're off to the races. Me, I just operate on the premise that everyone is out to get me and I'm good to go. For the more scientific amongst you, there's always a Faraday option. What? You don't have a Faraday Cage[1] in your house? You're going to need more tinfoil (figure 1). :-)

Figure 1 – Tinfoil coupon

In all seriousness, Faraday, in the *toolsmith* context, is an integrated penetration-test environment (IPE); think of it as an IDE for penetration testing designed for distribution, indexation, and analysis of the generated data during the process of a security audit (pentest) conducted with multiple users. It was some years ago when we discussed them in *toolsmith*, but Raphael Mudge's Armitage[2] is a similar concept for Metasploit, while Dradis[3] provides information sharing for pentest teams.

Faraday now includes plug-in[4] support for over 40 tools, including some *toolsmith*[5] topics and favorites such as Openvas, BeEF Arachni, Skipfish, and ZAP.

The Faraday project offers a robust wiki and a number of demo videos[6] you should watch as well.

I pinged Federico Kirschbaum, Infobyte's CTO and project lead for Faraday. He stated that, as learned from doing security assessments, they always had the need to know what the results were from the tests performed by other team members. Sharing partial knowledge of target systems proved to be useful not only to avoid overlapping but also to reuse discoveries and build a complete picture. During penetration tests where the scope is quite large, it is common that a vulnerability detected in one part of the network can be exploited somewhere else as well. Faraday's purpose is to aid security professionals, and its development is driven by this desire to truly convert penetration testing into a community experience.

Federico also described their goal to provide an environment where all the data generated during a pentest can be transformed into meaningful, indexed information. Results can then be easily distributed between team members in real time without the need to change workflow or tools, allowing them to benefit from the shared knowledge. Pentesters use a lot of tools on a daily basis, and everybody has a "favorite" tool set, ranging from full-blown vulnerability scanners to in-house tools. Instead of trying to change the way people like to work, the team designed Faraday as a bridge that allows tools to work in a collaborative way. Faraday's plug-in engine currently supports more than 40 well-known tools and also provides an easy-to-use API to support custom tools.

Information persisted in Faraday can be queried, filtered, and exported to feed other tools. As an example, one could extract all hosts discovered running SSH in order to perform mass brute-force attacks or see which commands or tools have been executed.

Federico pointed out that Faraday wasn't built thinking only about pentesters. Project managers can also benefit from a central database containing several assessments at once while being able to easily see the progress of their teams and have the ability to export information to send status reports.

It was surprising to the Infobytes team that many of the companies that use Faraday today are pentest clients rather than the actual pentest consultant. This is further indication of why it is always useful to have a repository of penetration test results whether they be internal or through outside vendors.

Faraday comes in three flavors: community, professional, and corporate. All of the features mentioned above are available in our community version, which is open source. I tested the community version for this effort as it is free.

Federico, in closing, pointed out that one of the main features in the commercial version is the ability to export reports for

1 http://www.thesurvivalistblog.net/build-your-own-faraday-cage-heres-how/.
2 http://holisticinfosec.org/toolsmith/pdf/january2011.pdf.
3 http://holisticinfosec.org/toolsmith/pdf/april2010.pdf.
4 https://github.com/infobyte/faraday/wiki/Plugin-List.
5 http://holisticinfosec.org/df/toolsmith.
6 https://github.com/infobyte/faraday/wiki/Demos.

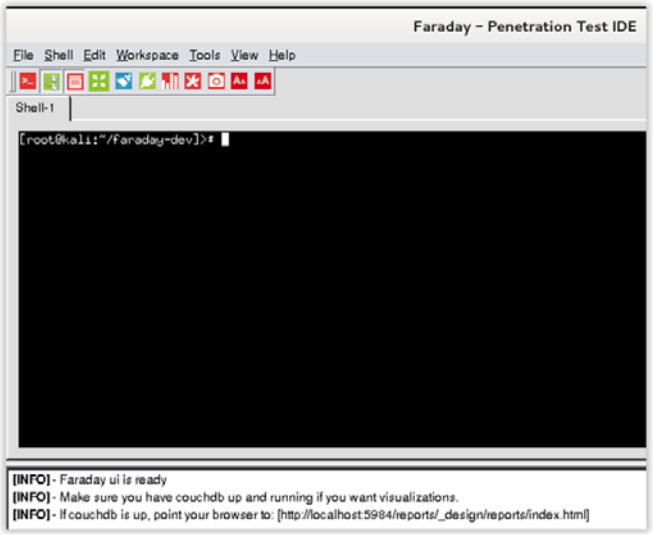

Figure 2 – Initial Faraday GUI QT

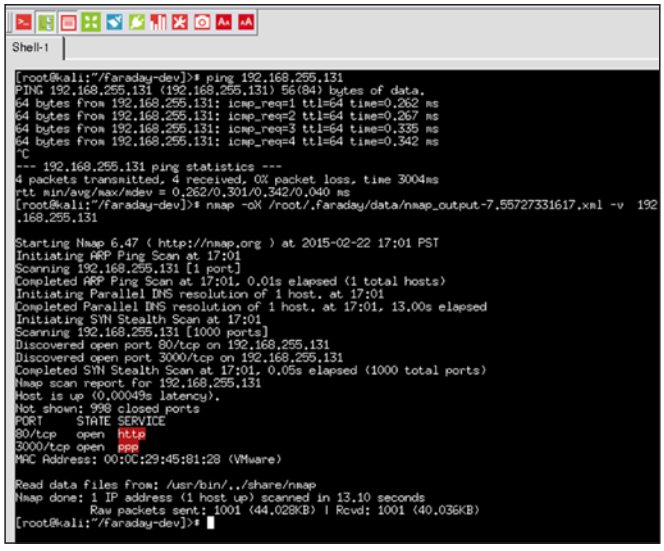

Figure 3 – Preliminary Faraday results

MS Word containing all the vulnerabilities, graphs, and progress status. This makes reporting, a pentester's bane (painful, uncomfortable, unnatural even), into a one-click operation that can be executed by any team member at any time. See the product comparison page for more features and details for versions,[7] based on your budget and needs.

Faraday preparation

The easiest way to run Faraday, in my opinion, is from Kali. This is a good time to mention that Kali 1.1.0 has been available as of 9 FEB 2015. If you haven't yet upgraded, I recommend doing so soon.

At the Kali terminal prompt, execute:

```
git clone https://github.com/infobyte/faraday.git
faraday-dev
cd faraday-dev
./install.sh
```

The installer will download and install dependencies, but you'll need to tweak CouchDB to make use of the beautiful HTML5 reporting interface. Use vim or Leafpad to edit */etc/couchdb/local.ini* and uncomment (remove semicolon) for *port* and *bind_address* on lines 11 and 12. You may want to use the Kali instance's IP address, rather than the loopback address to allow remote connections (other users). You can also change the port to your liking. Then restart the CouchDB service with `service couchdb restart`. You can manipulate SSL and authentication mechanisms in local.ini as well. Now issue `./faraday.py -d`. I recommend running with –d as it gives you all the debug content in the logging console. The service will start; the QT GUI will spawn (figure 2); and if all goes well, you'll receive an INFO message telling you where to point your browser for the CouchDB reporting interface. Note that there are limitations specific to reporting in the community version as compared to its commercial peers.

Fragging with Faraday

The first thing you should do in the Faraday UI is create a workspace: *Workspace | Create*. Be sure to save it as CouchDB as opposed to FS. I didn't enable replication as I worked alone for this assessment.

Shockingly, I named mine *toolsmith*. Explore the plug-ins available thereafter with either *Tools | Plugin* or use the *Plugin* button, fourth from the right on the toolbar. I started my assessment exercise against a vulnerable virtual machine (192.168.255.131) with a quick ping and nmap via the Faraday shell (figure 3). To ensure the default visualizations for Top Services and Top Host populated in the Faraday Dashboard, I also scanned a couple of my gateways.

As we can see in figure 3, our target host appears to be listening on port 80, indicating a web server—a great time to utilize a web application scanner. Some tools such as the commercial Burpsuite Pro have a Faraday plug-in for direct integration; you can still make use of free Burpsuite data, as well as results from the likes of free and fabulous OWASP ZAP. To do so, conduct a scan and save the results as XML to the applicable workspace directory, *~/.faraday/report/toolsmith* in my case. The results become evident when you right-click the target host in the Host Tree as seen in figure 4.

We can see as we scroll through findings we've discovered a SQL-injection vulnerability; no better time to use sqlmap, also supported by Faraday. Via the Faraday shell I ran the following, based on my understanding of the target apps discovered with ZAP.

To enumerate the databases:

```
sqlmap -u 'http://192.168.255.134/mutillidae/index.
php?page=user-info.php&username=admin&password=&
user-info-php-submit-button=View+Account+Details'
–dbs
```

To enumerate the tables present in the Joomla database:

```
sqlmap -u 'http://192.168.255.134/mutillidae/index.
php?page=user-info.php&username=admin&password=&
```

7 https://github.com/infobyte/faraday/wiki/Version-comparision.

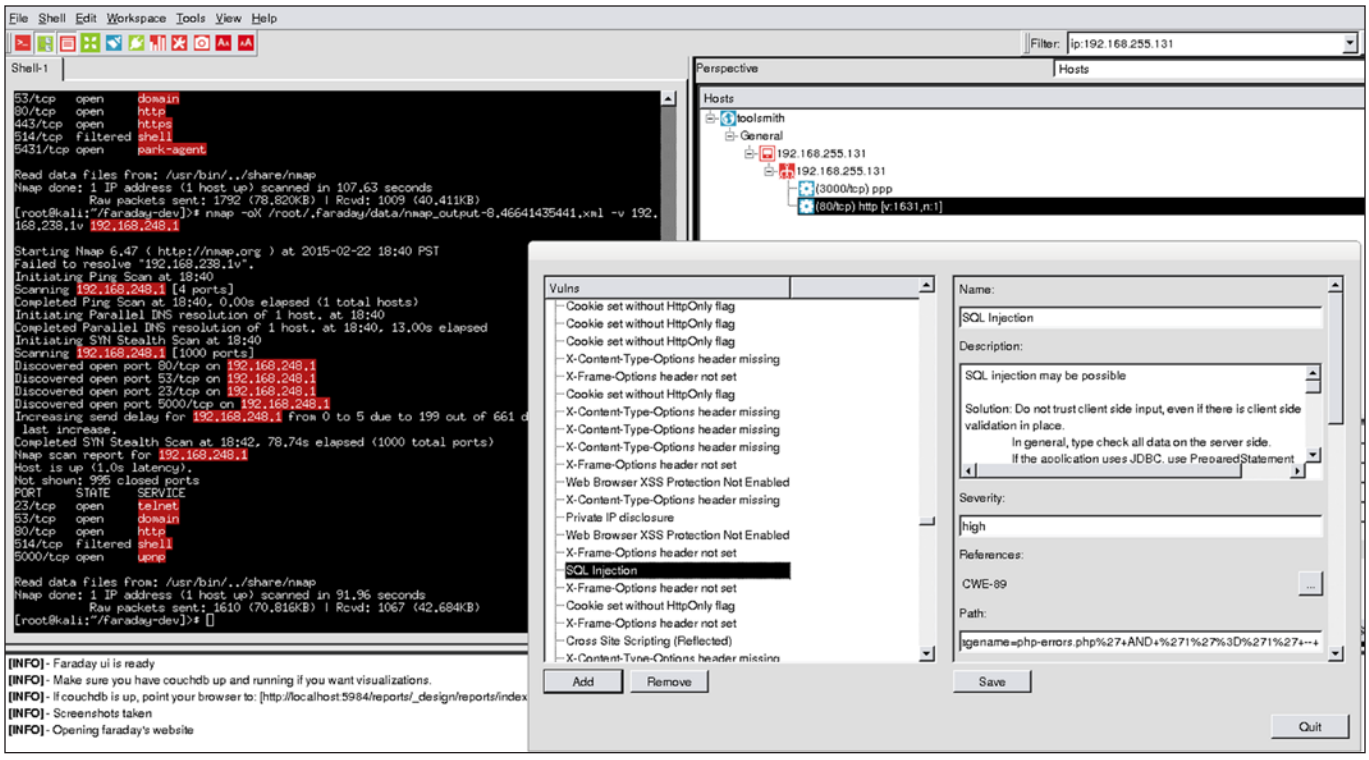

Figure 4 – Faraday incorporates OWASP ZAP results

```
user-info-php-submit-button=View+Account+Details'
-D joomla –tables
```

To dump the users from the Joomla database:

```
sqlmap -u 'http://192.168.255.134/mutillidae/index.
php?page=user-info.php&username=admin&password=&
user-info-php-submit-button=View+Account+Details'
--dump -D joomla -T j25_users
```

Unfortunately, late in the game as this was being written, we discovered a change in sqlmap behavior that causes some misses for the Faraday sqlmap plug-in, preventing sqlmap data from being populated in the CouchDB and thus the Faraday host tree. Federico immediately noted the issue and was issuing a patch as I was writing; by the time you read this you'll likely be working with an updated version. I love sqlmap so much though and wanted you to see the Faraday integration. Figure 5 gives you a general sense of the Faraday GUI accommodating all this sqlmap mayhem.

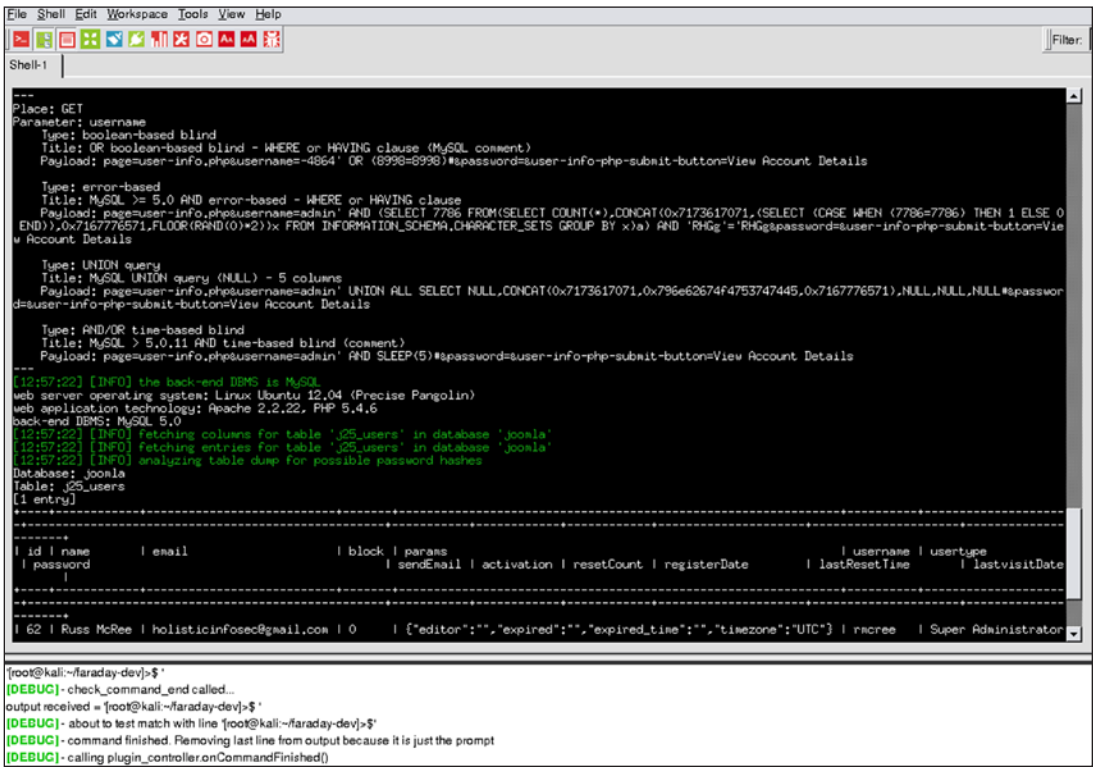

Figure 5 – Faraday shell and sqlmap

That being said, here's where all the real Faraday superpowers kick in. You've enumerated, assessed, and even exploited, now to see some truly beautified HTML5 results. Per

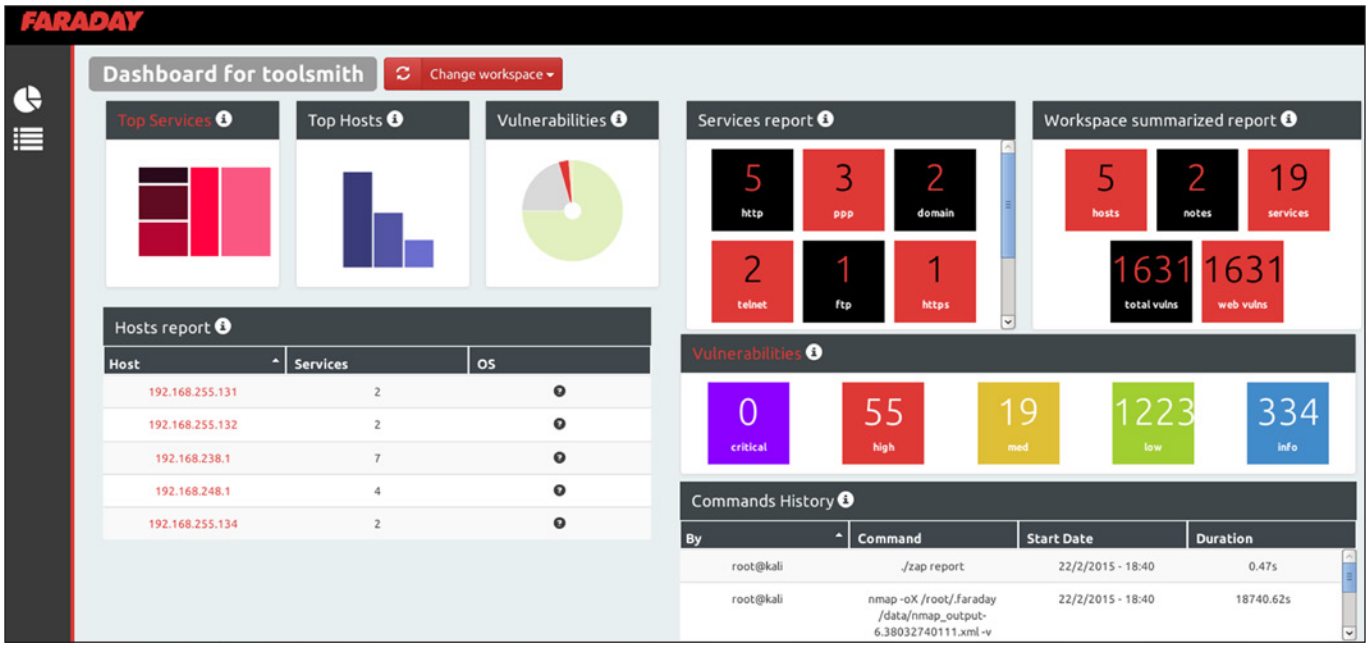

Figure 6 – Faraday dashboard

figure 6; the Faraday dashboard is literally one of the most attractive I've ever seen and includes different workspace views, hover-over functionality, and host drilldown.

There's also the status-report view, which really should speak for itself but allows you really flexible filtering as seen in figure 7.

Those pentesters and pentest PMs who are looking for a data-management solution should now be fully inspired to check out Faraday in its various versions and support levels. It's an exciting tool for a critical cause.

In conclusion

Faraday is a project that benefits from your feedback, feature suggestions, bug reports, and general support. They're an engaged team with a uniquely specialized approach to problem solving for the red team cause, and I look forward to future releases and updates. I know more than one penetration testing team to whom I will strongly suggest Faraday consideration.

Ping me via email or Twitter if you have questions (russ at holisticinfosec dot org or @holisticinfosec).

Cheers…until next month.

Acknowledgements

—Federico Kirschbaum (@fede_k), Faraday (@faradaysec) project lead, CTO Infobyte LLC (@infobytesec)

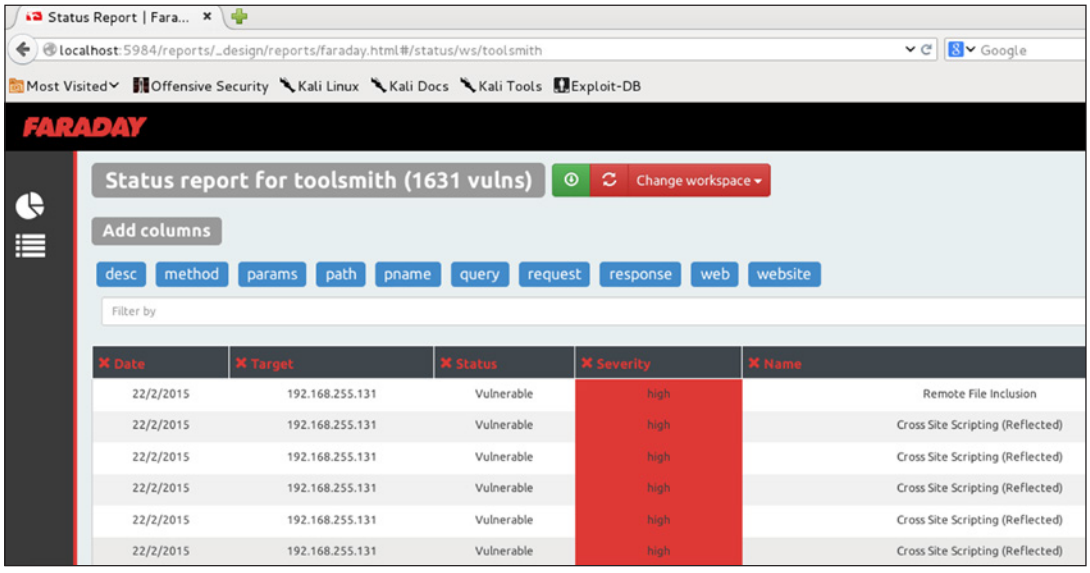

Figure 7 – Faraday Status

About the Author

Russ McRee manages the Threat Intelligence & Engineering team 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. Reach him at russ at holisticinfosec dot org or @holisticinfosec.