# SpiderFoot

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

### Prerequisites/dependencies

Python 2.7 if running on *nix as well as M2Crypto, CherryPy, netaddr, dnspython, and Mako modules

Windows version comes as a pre-packaged executable, no dependencies

All good penetration tests and threat assessments should be initiated with what you've seen referred to in *toolsmith*[1] as OSINT,[2] or open source intelligence gathering. This practice contributes greatly to collecting a useful list of targets of opportunity. One key element to remember though; the bad guys are conducting this same activity against you and your Internet-facing assets too. It's probably best then that you develop your own OSINT practice so you can find the information you may not wish, or even know, you are exposing. Steve Micallef's SpiderFoot is another tool in the arsenal specific to this cause. You may already be aware that the four phases of a web application security assessment, as defined using the SamuraiWTF distribution, are recon, mapping, discovery, and exploitation. The SANS GIAC Certified Web Application Penetration Tester (GWAPT) curriculum follows suit, given that Secure Idea's Kevin Johnson contributed heavily (developed) to both. SpiderFoot nicely blends both recon and mapping as part of its feature set.

As we consider legal, privacy, and ethics issues for the March *ISSA Journal*, OSINT and reconnaissance become interesting and related topics. I have, on more than one occasion, discovered very damaging data via OSINT tactics that, if in the wrong hands, could have been very damaging. When you consider findings of this nature with regard to ethics and legality, you may find yourself in an immediate quandary. Are you obligated to report findings that you know could cause harm to the target if left unmitigated? What if during your analysis you come into possession of classified or proprietary information that having in your possession could create legal challenges for you? Imagine findings of this caliber and it becomes easy to recognize why you should always conduct intelligence gathering and footprinting on your own interests before the wrong people do it for you. SpiderFoot, as a tool for just such purposes, allows you to understand "as much as possible about a given target in order to perform a more complete security penetration test." For large networks, this can be a daunting task, and SpiderFoot automates this process significantly, allowing penetration testers to focus their efforts on security testing itself.

Steve provided us with some SpiderFoot history as well as insight on what he finds useful and interesting. He originally wrote SpiderFoot as a C# .NET application in 2005, purely as an exercise to learn C#, having been inspired by BiDiBLAH's developers from Sensepost (who went on to create Maltego), thinking he could make a lighter open source version. For seven years that was Steve's first and only release until he decided to resume development again in 2012. His work on next generation versions have led SpiderFoot to be cross platform (Python), far more extensible, functional, and with a much nicer user interface (UI).

Steve's current challenge with SpiderFoot is deciding what cool functionality to implement next; his to-do list is ever growing, and there are numerous features he'd love to extend it to include. He typically balances his time between UI/analysis functionality versus new checks to identify more items to aid the penetration tester. The aforementioned OSINT (open source intelligence) community also continues to produce new sources, which in turn inspire Steve to build new SpiderFoot checks.

He finds it interesting testing out a new module and actually finding insightful items out there on the Internet simply during the development process. Steve's favorite functionality at the moment is identifying owned netblocks and co-hosted sites. Owned netblocks indicates entire IP ranges that an organization owns, which enables penetration testers to more completely scan the perimeter of a target. Co-hosted sites shows you any websites on the same server as the target, which can also be revealing. If your target is hosted on the same server as sites identified as being malicious by the malicious site checker or the blacklist checker plug-in, it could potentially indicate that your target is hosted on a compromised server.

As you read this it's likely that the following planned enhancements are available in SpiderFoot or will be soon:

- 2.1.2 (late February)
  - SOCKS proxy support
  - Real-time scan progress viewer
  - Identify scan quality impacting issue

1 http://holisticinfosec.org/toolsmith-issa-mainmenu-26.
2 http://holisticinfosec.blogspot.com/search?q=osint.



©2014 ISSA • www.issa.org • editor@issa.org • Permission for author use only.

- Autoshun (www.autoshun.org) lookup as part of malicious checks
- SANS (isc.sans.edu) lookup as part of malicious checks (queue the Austin Powers voice: "Yeah, baby!")
- Update GeoIP checker

- 2.1.3 Release: Late March
  - VirusTotal, SHODAN, Facebook, Xing, Pastebin, and Github plug-ins

SpiderFoot is a great project with a strong development roadmap, so let's get down to business and explore.

## Quick installation notes

Windows installation is an absolute no brainer; download the package,[3] unpack it, execute `sf.exe`, and browse to http://127.0.0.1:5001. All dependencies are met, including a standalone Python interpreter, so you may find this option optimal.

Linux (I installed it on SamuraiWTF) users need to settle a few dependencies easily solved with the following few steps that assume pip is already installed:

```
sudo apt-get install swig
sudo pip install mako cherrypy
netaddr M2Crypto dnspython
git clone https://github.com/
smicallef/spiderfoot.git
cd spiderfoot/
sudo python ./sf.py 0.0.0.0:9999
```

The last line indicates that you'd like SpiderFoot to bind to all addresses (including localhost) and listen on port 9999. You can define your preferred port or just accept default if undefined (5001). Steve reminds us on his installation page[4] to be cautious regarding exposing SpiderFoot to hostile networks (Intranet, security conference wireless) given that there is currently no authentication scheme.

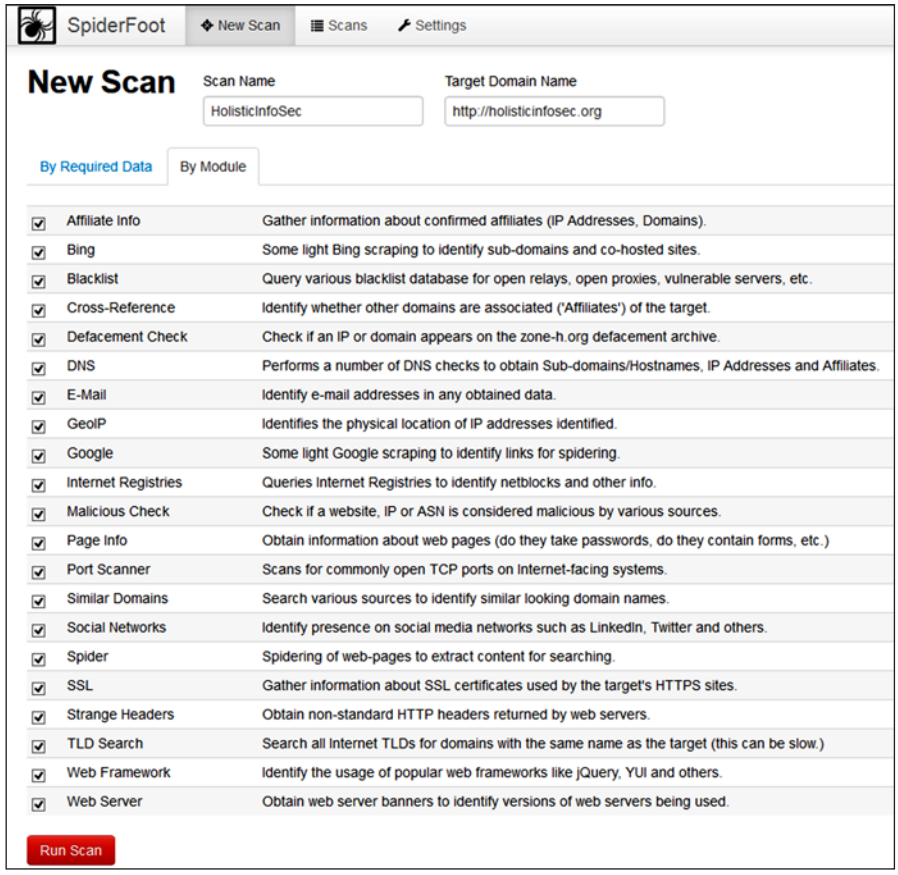


**Figure 1 – Kicking off a new scan with SpiderFoot**

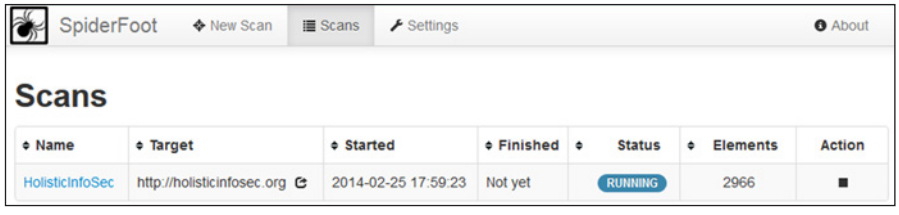


**Figure 2 – SpiderFoot Scans view**

## SpiderFoot unleashed

The SpiderFoot UI is, how shall I say, incredibly simple, intuitive, and obvious even. To start a scan…wait for it…select *New Scan*. Figure 1 represents a scan being kicked off on my domain (don't do it) as defined by the By Module view.

If you wish to more granularly define your scans, select the *By Required Data* view (default) then pick and choose your preferred data points, including elements such as malicious affiliations, IP data, URL analysis, SSL certificate information, affiliate details, and many other records. You should then be treated to a success message. Scans results are stored in a SQLite DB so over time you'll likely build up a collection if you don't purge. Under the *Scans* tab, as seen in figure 2, you can click the scan in the *Name* column of the table view and review results. You'll also note status here and can halt the scan if need be. I imagine the real-time scan progress viewer will show itself here in the near future as well.

If need be (default settings work quite well), you can tune the actual scan configuration as well via *Settings*, with attention to how you'd like to tune storage, search engines, port scanning, spidering, TLD searches (see figure 3), amongst others.

When my scan completed, with default settings and all checks enabled, the results included 11360 elements. For you data miners, metrics minions, and hosting harvesters, you can export the results to CSV (see figure 4) and filter by findings type and module, or your preferred data pivot.

3 http://sourceforge.net/projects/spiderfoot/files/SpiderFoot-2.1.1-w32.zip/download.

4 https://github.com/smicallef/spiderfoot/wiki/Installing.

©2014 ISSA • www.issa.org • editor@issa.org • Permission for author use only.

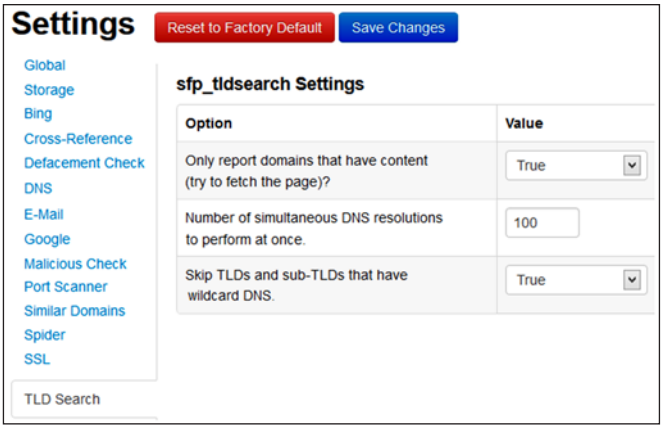


**Figure 3 – SpiderFoot Settings view**

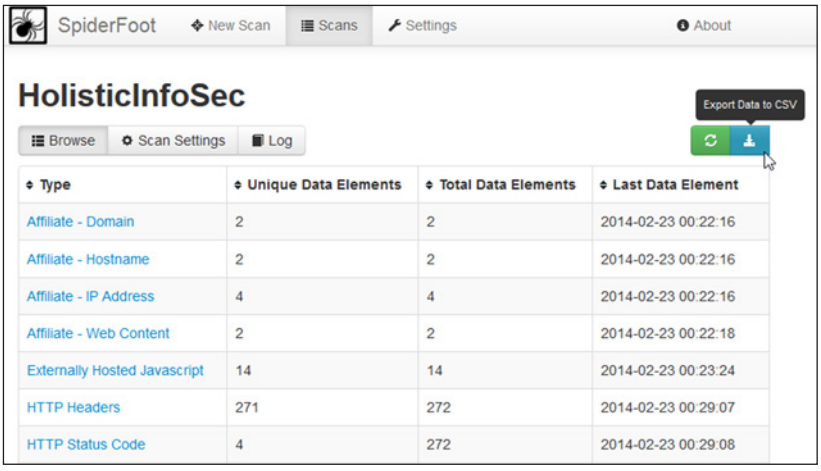


**Figure 4 – SpiderFoot results and export functionality**

As I navigated all the results, I was intrigued to find a hit for *URL (Uses Flash)* simply because I didn't recall any Flash features on my site. I immediately chuckled when I reviewed the result as it was specific to a Flash video[5] I'd created for the 2008 ISSA Northwest Regional Conference wherein I ripped on the now defunct Hacker Safe trustmark for indicating that their customer's sites were "hacker safe" when, in fact, they were not. Oh, the good old days.

Want to visualize your results? No problem, you can choose from a bubble view of data elements or the discovery path. Figure 5 represents the discovery path for *Social Media Presence* findings. Hover over each entity for details specific to initial target type, the source module, and the related result.

SpiderFoot will absolutely uncover nuggets you may have long forgotten about and may want to remove as they are potentially vulnerable (outdated plugins, modules, etc.) or unnecessarily/unintentionally exposed. I found an old dashboard I'd built by hand eons ago with long dead extenal JavaScript calls that had no business still being available. "Be gone!" I said. That is what SpiderFoot is all about. Add it to the tool collection for penetration tests and OSINT expeditions; you won't be disappointed.

## In conclusion

Steve Micallef's SpiderFoot is functionally simple but feature rich and getting better all the time as it is well built and maintained. Follow @binarypool on Twitter and keep an eye out for timely and regular releases.

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

Cheers…until next month.

### Acknowledgements

—Steve Micallef (@binarypool), Spiderfoot author

## 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. 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.*

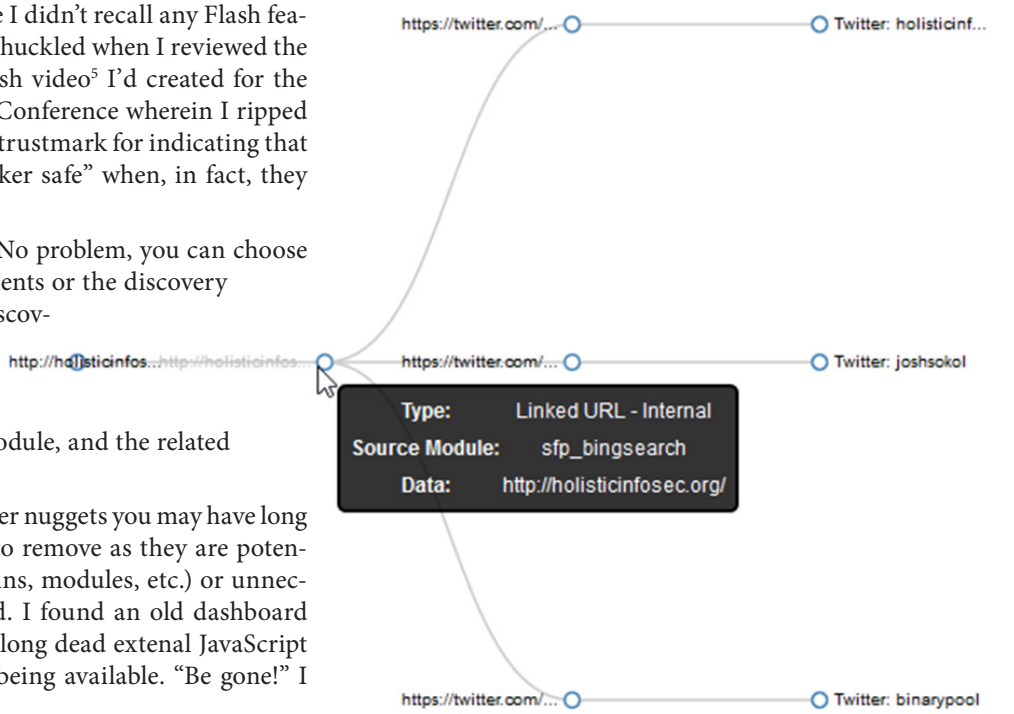


**Figure 5 – SpiderFoot visualizes a discovery path**

5 http://holisticinfosec.org/video/HS_ISSA/ISSA_Regional_HackerSafe.html.



©2014 ISSA • www.issa.org • editor@issa.org • Permission for author use only.