SearchDiggity: Dig Before They Do

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

Prerequisites

Windows .NET Framework

I've been conducting quite a bit of open-source intelligence gathering (OSINT) recently as part of a variety of engagements and realized I hadn't discussed the subject since we last reviewed FOCA[1] in March 2011 or "Search Engine Security Auditing"[2] in June 2007. I'd recently had a few hits on my feed reader, and at least via one mailing lists, regarding SearchDiggity from Fran Brown and Rob Ragan of Stach & Liu. They'd recently presented *Pulp Google Hacking* at the 4th Annual InfoSec Summit at ISSA Los Angeles as well as *Tenacious Diggity* at DEFCON 20, and the content certainly piqued my interest. One quick look at the framework and all its features and I was immediately intrigued.

At first glance you note similarities to Wikto and FOCA, given Search Diggity's use of the Google Hacking Database and Shodan. This is no small irony as this team has taken point on rejuvenating the art of the search engine hack. In Fran's *InformationWeek* report, "Using Google to Find Vulnerabilities In Your IT Environment,"[3] he discusses *toolsmith* favorites FOCA, Maltego, and Shodan amongst others. I'll paraphrase Fran from this March 2012 whitepaper to frame why using tools such as SearchDiggity and others in the Diggity arsenal is so important. Use these same methods to find flaws before the bad guys do; these methods use search engines such as Google and Bing to identify vulnerabilities in your applications, systems, and services, allowing you to fix them before they can be exploited.

Fran and Rob's work has even hit mainstream media with the likes of NotInMyBackyard (included in SearchDiggity) achieving coverage in *USA Today*.[4] Suffice it to say that downloads from the Google Hacking Diggity Project[5] pages jumped by 45,000 almost immediately, fueled largely by non-security consumers looking to discover any sensitive data leaks related to themselves or their organizations. A nice problem to have for the pair from Stach & Liu and one Fran addressed with a blogpost to provide a quick intro to NotInMyBackYard Diggity,[6] to be discussed in more detail later in this article.

I reached out to Fran and Rob rather late in this month's writing process and am indebted to them as they kindly accommodated me with a number of resources as well a few minutes for questions via telephone. There are Diggity-related videos and tool screenshots[7] as well as all the presentations[8] the team has given in the last few years. The SearchDiggity team is most proud of their latest additions to the toolset, including NotInMyBackyard and PortScan. Keep in mind that, like so many tools discussed in *toolsmith*, SeachDiggity and its various elements were written to accommodate the needs of the developers during their own penetration tests and assessments. No cached data is safe from the Diggity Duo's next generation search engine hacking arsenal and all their tools are free for download and use.

Installing SearchDiggity

SearchDiggity installation is point-and-click simple after downloading the installation package, but there are few recommendations for your consideration. The default installation path is `C:\Program Files (x86) \SearchDiggity`, but consider using a non-system drive as an installation target to ensure no permissions anomalies; I installed in `D:\tools\SearchDiggity`. SearchDiggity writes results files to `DiggityDownloads` (I set `D:\tools\DiggityDownloads` under *Options* → *Settings* → *General*) and will need permission to its root in order to Update Query Definitions (search strings, Google/Bing Dorks).

Using SearchDiggity

I started my review of SearchDiggity capabilities with the Bing Hacking Database (BHDB) under the Bing tab and utilizing the menu referred to as *BHDBv2NEW* as seen in figure 1 on the next page.

As with any tool, optimization of your scan settings for your target before you start the scan run is highly recommended. Given that my site is not an Adobe Coldfusion offering, there's really no need to look for CFIDE references, right? Ditto for

1 http://holisticinfosec.org/toolsmith/pdf/march2011.pdf.

2 http://holisticinfosec.org/toolsmith/docs/june2007.pdf.

3 http://reports.informationweek.com/abstract/21/8703/Security/strategy-using-google-to-find-vulnerabilities.html.

4 http://www.usatoday.com/money/industries/technology/story/2012-08-01/online-consumer-privacy-tool/56719894/1.

5 http://www.stachliu.com/resources/tools/google-hacking-diggity-project/.

6 http://www.stachliu.com/2012/08/quick-intro-to-notinmybackyard-diggity/.

7 http://www.stachliu.com/resources/tools/google-hacking-diggity-project/diggity-media-gallery/.

8 http://www.stachliu.com/resources/tools/google-hacking-diggity-project/presentation-slides/.

Figure 1 – BHDB analysis of HolisticInfosec.org

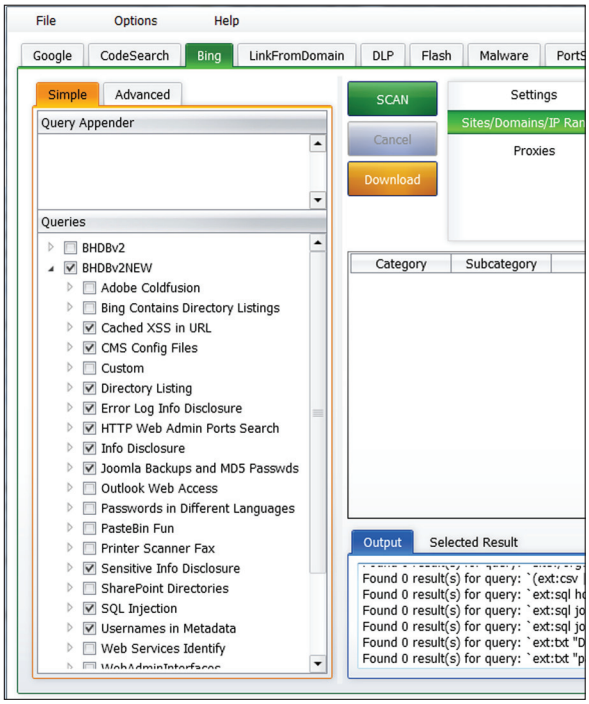
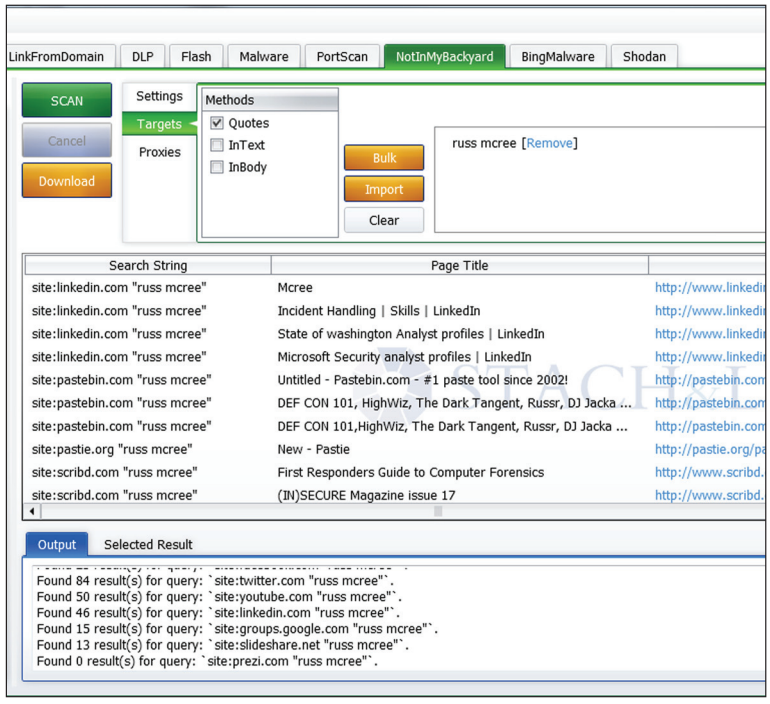


Figure 2 – NotInMyBackyard flushes out results



Outlook Web Access or SharePoint, but CMS Config Files with XSS and SQL *injection instreamset* options are definitely in order. Good news, no significant findings were noted using my domain as the target.

NotInMyBackyard is a recent addition to SearchDiggity for which the team has garnered a lot of deserved attention, and as such we'll explore it here. I used my name as my primary search parameter and configured *Methods* to include *Quotes*, and set *Locations* to include:

- Cloud Storage (Dropbox, Google Docs, Microsoft Skydrive, Amazon AWS)
- Document Sharing (scribd.com, 4shared.com, issuu.com, docstoc.com, wepapers.com)
- Pastebin (pastebin.com, snipt.org, drupalbin.com, paste.ubuntu.com, tinypaste.com, paste2.org, codepad.org, dpaste.com, pastie.org, pastebin.mozilla.org)
- Social (Facebook, Twitter, YouTube, LinkedIn)
- Forums (groups.google.com)
- Public presentations charts graphs videos (Slideshare, Prezi, present.me, Gliffy, Vimeo, Dailymotion, Metacafe)

You can opt to set additional parameters such as *Extensions* for document types including all versions of Microsoft Office, PDF, CSV, TXT, database types including MS-SQL and Access, backup, logs, and config files, as well as test and script files. My favorites (utilized in a separate run) are the financial file options including Quicken and QuickBooks data files and QuickBooks backup files. Finally, there are a number of granular keyword selections to narrow your query results that might include your patient records, places of birth, or your name in a data dump. This is extremely useful when trying to determine if your email address, as associated with one of your primary accounts, has been accumulated in a data dump posted to a Pastebin-like offering. Just keep in mind, the more options you select the longer your query run will take. I typically carve my searches up in specific categories then export the results to a file named for the category.

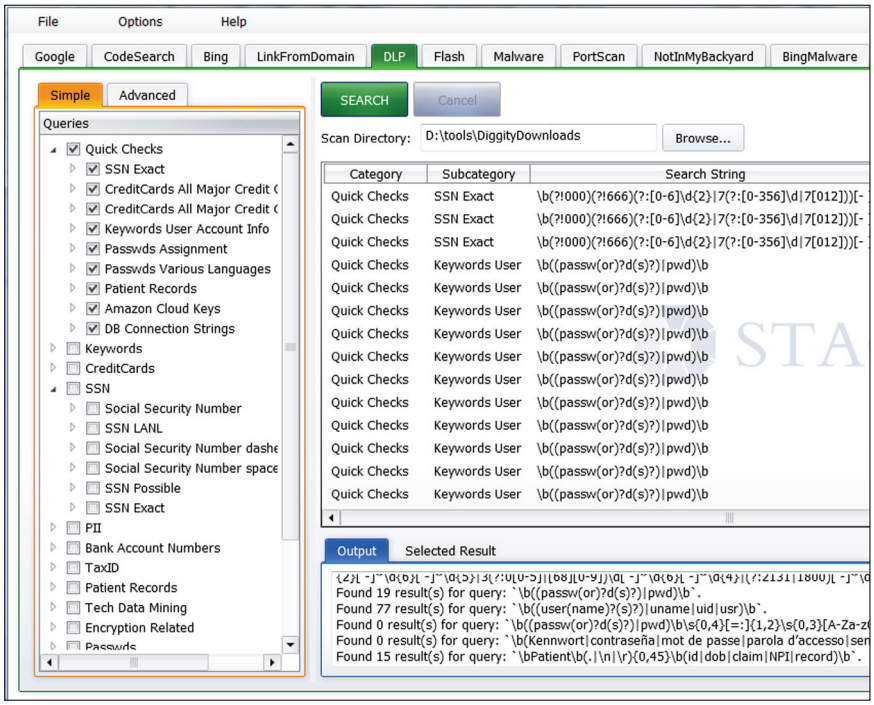
As seen in figure 2, NotInMyBackyard reveals all available query results in a clean, legible manner that includes hyperlinks to the referenced results, allowing you to validate the findings.

I found that my search, as configured, was more enlightening specific to all the copies of my material posted to other sites without my permission. It was also interesting to see where articles and presentation material were cited in academic material. Imagine using your organizational domain name, and specific keywords and accounts to discover what's exposed to the evildoers conducting the same activity.

You can focus similar activity with more attention to the enterprise mindset utilizing SearchDiggity's DLP offerings. First conduct a Google or Bing run against a domain of interest using the *DLPDiggity Initial* selection. Once the query run is complete, highlight all the files (CTRL-A works well), and click the download button. This will download all the files to the download directory you configured, populating it with files discovered using DLPDiggity Initial, against which you can then apply the full DLP menu. I did as described against a target that shall remain unnamed and found either valid findings or sample/example data that matched the search regex explicitly as seen in figure 3.

I only used the *Quick Checks* set here. When you contemplate the likes of database connection strings, bank account

Figure 3 – Data Leak Prevention with SearchDiggity



numbers, and encryption-related findings, coupled with the requisite credit cards, SSNs, and other PII, it becomes immediately apparent how powerful this tool is for both prevention and discovery during the reconnaissance phase of a penetration test.

I'll cover one more SearchDiggity component, but as is usually the case with *toolsmith* topics there is much about the tool du jour that remains unsaid. Be sure to check out the SearchDiggity Shodan and PortScan offerings on your own. I'm always particularly interested in Flash-related FAIL findings and SearchDiggity won't disappoint here either. Start with a Google or Bing search against a target domain with *FlashDiggity Initial* enabled. Much as noted with the DLP feature, after discovery SearchDiggity will download the SWF files it identifies with FlashDiggity Initial. As an example I ran this configuration without a domain specified. By default, for a Google search, 70 results per query will be returned. Suffice it to say that with the three specific queries defined in FlashDiggity Initial searches, I was quickly treated to 210 results which I then opted to download. I switched over the Flash menu and for real s's and g's (work that one out on your own ☺) enabled all options. Figure 4 exemplifies (anonymously) just how concerning certain Flash implementations may be, particularly when utilized for administrative functions and authentication.

FlashDiggity decompiles the downloaded SWF files with Flare and stores the resulting `.flr` file in the download directory for your review. It should go without saying that flaw enumeration becomes all that much easier. As an example, FlashDiggity's getURL XSS detection discovered the following using `geturl\(.*(_root\.|_level0\.|_global\.).*\)` as its regex logic:

```
this.getURL('mailto:' + _global.escape(this.
decodeEmailAddr(v2.emladdr)) + '?subject=' + _
global.escape(v2.emlsubj) + '&body=' + _global.
escape(this.getEmailContent()));
```

This snippet makes for interesting analysis. Risks associated with *getURL* are well documented, but the global escape may mitigate the issue. That said, the Flash file was created with Techsmith Camtasia in January 2009, and an XSS vulnerability was reported in October 2009 regarding SWF files created with Camtasia Studio. Yet, SWF files hosted on TechSmith's Screencast service were not vulnerable and more than one reference to Screencast was noted in the decompiled `.flr` file. With one FlashDiggity search, we were able to learn a great deal about potentially flawed Flash files subject to possible exploit.

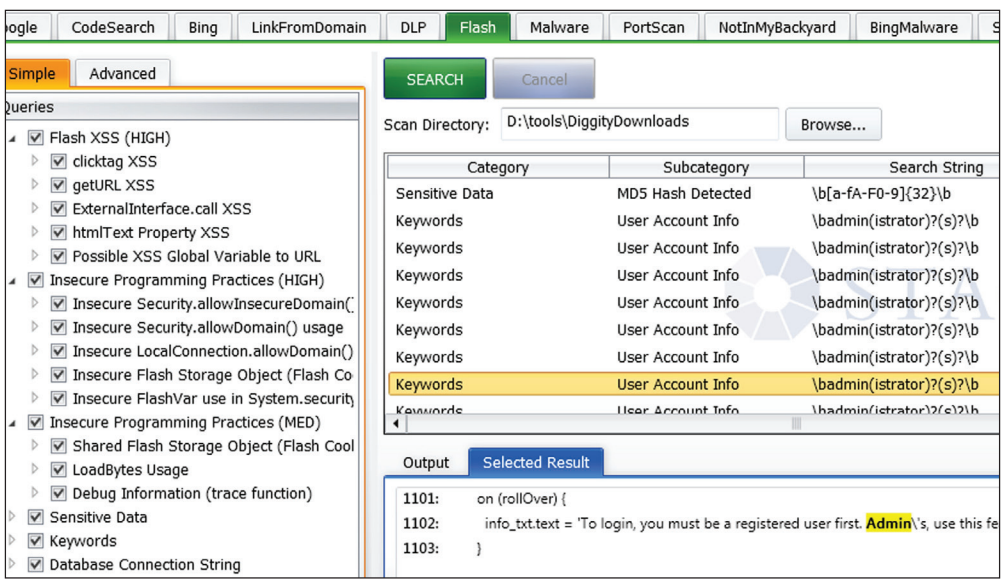


Figure 4 – Find bad Flash with SearchDiggity

And we didn't even touch SearchDiggity's malware analysis feature set.

In conclusion

As always I'll remind you, please use SearchDiggity for good, not evil. Incorporating its use as part of your organizational defensive tactics is a worthy effort. Keep in mind that you can also leverage this logic as part of Google Hacking Diggity Defense Tools, including Alert and Monitoring RSS feeds. Configure them with your specific and desired organizational parameters and enjoy real-time alerting and

monitoring via your RSS feed reader. For those of you defending Internet-facing SharePoint implementations, you'll definitely want to check out the SharePoint Diggity Hacking Project too.

Enjoy this tool arsenal from Stach & Liu's Dynamic Duo; they'd love to hear from you with kudos, constructive criticism, and feature requests via diggity at stachliu.com.

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

Cheers…until next month.

Acknowledgements

—Francis Brown and Rob Ragan, Managing Partners, Stach & Liu, Google Hacking Diggity project leads.

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.