

Use Dorkbot for Automated Vulnerability Discovery

- BY **BARROW**

- 10/14/2017 3:29 PM

- 10/17/2017 12:25 AM

If you need to scan a large number of domains for a specific web app vulnerability, Dorkbot may be the tool for you. Dorkbot uses search engines to locate dorks and then scan potentially vulnerable apps with a scanner module.

This tool is useful if you're managing a large number of hosts and aren't sure what may be vulnerable and what may not. It's also useful if you're a black hat looking to compromise as many machines as possible in a short time, not that we condone any black hattery here.

Before we get started, I'd like to explain the concept of a dork a little bit further. Dorks are a way of using search engines to locate vulnerable web apps. If you're thinking "that's just Google hacking," you're correct. They are essentially the same thing, though Google hacking generally has fewer negative connotations.

Essentially, when we use dorks, the goal is to search out a vulnerable application and either note it or attempt to exploit it. The internet is a big place, and if an attacker's goal is simply to amass a collection of vulnerable machines, Google dorks are the first place to start.

Don't Miss: [How to Hack Google Dorks](#)

This style of mass-vulnerability scanning is advantageous for a few reasons: Finding targets is easy, and the search engine does the work for you. Exploiting the targets is also easy. If you've done some research, you know exactly what vulnerability you are looking to exploit. This means you have the exploit code and you've tested it.

This makes the entire attack on the vulnerable host much easier. Rather than encountering a host and going through the entire methodology of an attacking something unknown, the vulnerable hosts, in this case, come to you.

With that covered, let's get started discovering with Dorkbot.

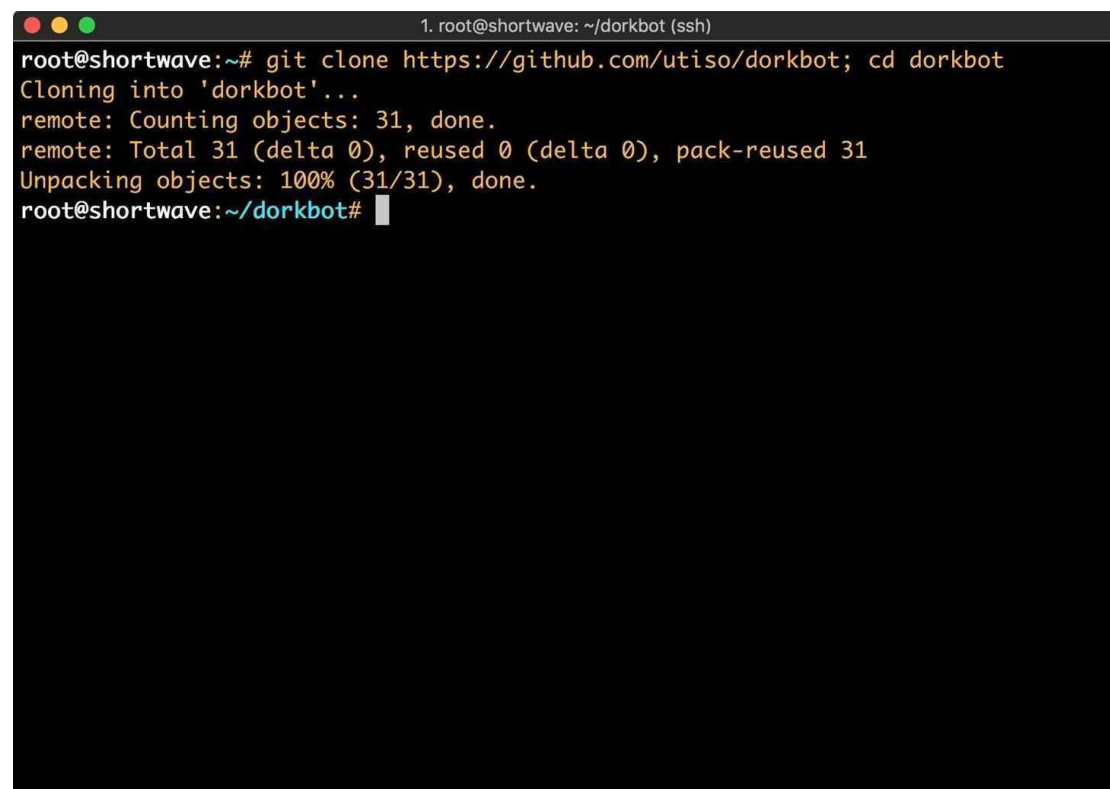
Step 1 Install Dorkbot on Kali

For this tutorial, I will be using [Kali Linux](#), logged in as the root user. Before we get started, we should probably update our system. Run **apt** in your terminal emulator to do so with the following commands.

```
apt update && apt upgrade
```

Once that command completes, we can start installing Dorkbot. The first thing to do is pull the repository off of GitHub, using **git** in your favorite terminal emulator.

```
git clone https://github.com/utiso/dorkbot; cd dorkbot
```

A terminal window titled "1. root@shortwave: ~/dorkbot (ssh)" shows the execution of the command "git clone https://github.com/utiso/dorkbot; cd dorkbot". The output indicates that the repository was successfully cloned into the 'dorkbot' directory. The terminal text is as follows:

```
root@shortwave:~# git clone https://github.com/utiso/dorkbot; cd dorkbot
Cloning into 'dorkbot'...
remote: Counting objects: 31, done.
remote: Total 31 (delta 0), reused 0 (delta 0), pack-reused 31
Unpacking objects: 100% (31/31), done.
root@shortwave:~/dorkbot#
```

Next, you will need to download and install dependencies. The first of these is [PhantomJS](#). We will download and extract it into the dorkbot/tools directory, and then rename the extracted folder to "phantomjs" with the following commands.

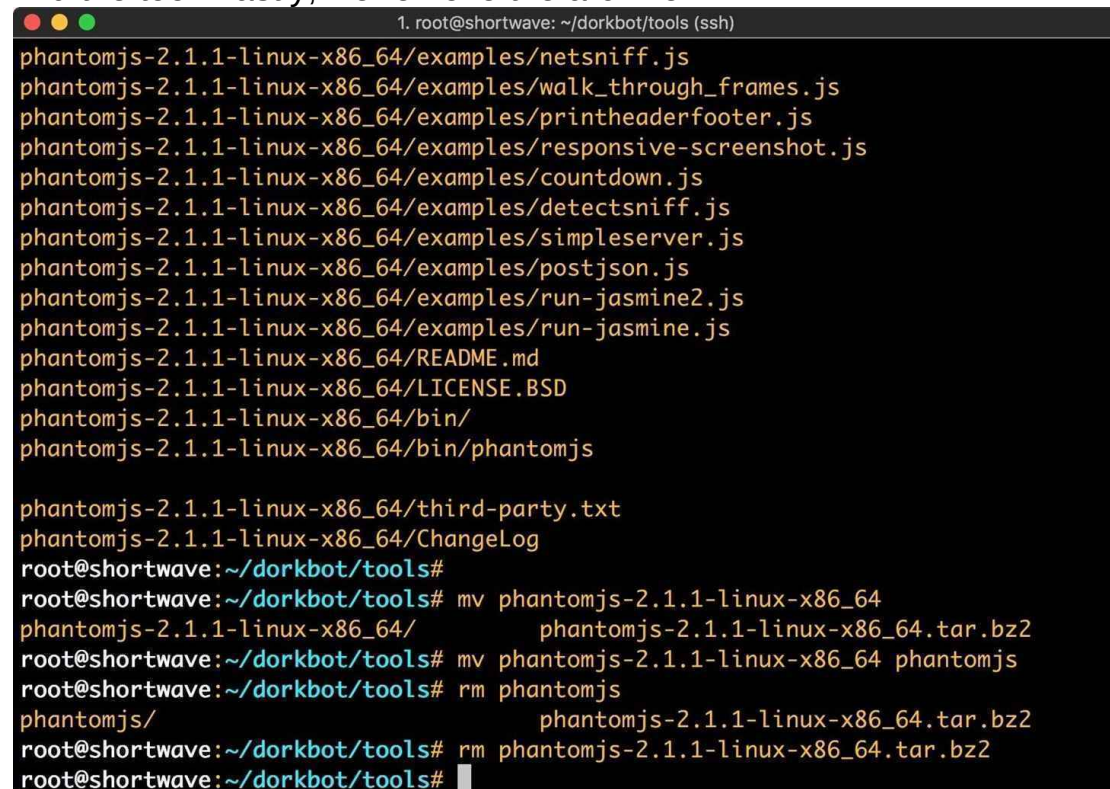
```
wget https://bitbucket.org/ariya/phantomjs/downloads/phantomjs-2.1.1-linux-x86\_64.tar.bz2
```

```
tar vxjf phantomjs-2.1.1-linux-x86_64.tar.bz2
```

```
mv phantomjs-2.1.1-linux-x86_64 phantomjs
```

```
rm phantomjs-2.1.1-linux-x86_64.tar.bz2
```

The URL in the **wget** command may change as PhantomJS is updated, you can always check the PhantomJS site for the most recent URL. The **tar** command extracts the PhantomJS archive, then we rename the directory with the "mv" command so that Dorkbot can find the tool. Lastly, we remove the archive.

A terminal window titled '1. root@shortwave: ~/dorkbot/tools (ssh)' showing the installation of PhantomJS. The terminal lists the contents of the extracted directory, including example JavaScript files, README, LICENSE, and the bin directory. It then shows the removal of the directory and the tar archive.

```
phantomjs-2.1.1-linux-x86_64/examples/netsniff.js
phantomjs-2.1.1-linux-x86_64/examples/walk_through_frames.js
phantomjs-2.1.1-linux-x86_64/examples/printheadefooter.js
phantomjs-2.1.1-linux-x86_64/examples/responsive-screenshot.js
phantomjs-2.1.1-linux-x86_64/examples/countdown.js
phantomjs-2.1.1-linux-x86_64/examples/detectsniff.js
phantomjs-2.1.1-linux-x86_64/examples/simpleserver.js
phantomjs-2.1.1-linux-x86_64/examples/postjson.js
phantomjs-2.1.1-linux-x86_64/examples/run-jasmine2.js
phantomjs-2.1.1-linux-x86_64/examples/run-jasmine.js
phantomjs-2.1.1-linux-x86_64/README.md
phantomjs-2.1.1-linux-x86_64/LICENSE.BSD
phantomjs-2.1.1-linux-x86_64/bin/
phantomjs-2.1.1-linux-x86_64/bin/phantomjs

phantomjs-2.1.1-linux-x86_64/third-party.txt
phantomjs-2.1.1-linux-x86_64/Changelog
root@shortwave:~/dorkbot/tools#
root@shortwave:~/dorkbot/tools# mv phantomjs-2.1.1-linux-x86_64
phantomjs-2.1.1-linux-x86_64/ phantomjs-2.1.1-linux-x86_64.tar.bz2
root@shortwave:~/dorkbot/tools# mv phantomjs-2.1.1-linux-x86_64 phantomjs
root@shortwave:~/dorkbot/tools# rm phantomjs
phantomjs/ phantomjs-2.1.1-linux-x86_64.tar.bz2
root@shortwave:~/dorkbot/tools# rm phantomjs-2.1.1-linux-x86_64.tar.bz2
root@shortwave:~/dorkbot/tools#
```

The next dependency that needs to be resolved is our scanner module. Dorkbot works with two different scanner modules, [Arachni](#) and [Wapiti](#). You will need to select one of these to use as your scanner. After testing with Wapiti, I found that it threw errors, so I settled on Arachni. To install it, run the following in a terminal window.

```
wget https://github.com/Arachni/arachni/releases/download/v1.5.1/arachni-1.5.1-0.5.12-linux-x86\_64.tar.gz
```

```
tar xzf arachni-1.5.1-0.5.12-linux-x86_64.tar.gz
```

```
mv arachni-1.5.1-0.5.12
```

```
rm arachni-1.5.1-0.5.12-linux-x86_64.tar.gz
```

```
1. root@shortwave: ~/dorkbot/tools/arachni (ssh)
equest&X-Amz-Date=20171006T201931Z&X-Amz-Expires=300&X-Amz-Signature=6c450c28cce
cd69f96d22b88d5eb463c18f23f7522819c9a3832216e70ef0ffc&X-Amz-SignedHeaders=host&a
ctor_id=0&response-content-disposition=attachment%3B%20filename%3Darachni-1.5.1-
0.5.12-linux-x86_64.tar.gz&response-content-type=application%2Foctet-stream
Resolving github-production-release-asset-2e65be.s3.amazonaws.com (github-produc
tion-release-asset-2e65be.s3.amazonaws.com)... 52.216.1.232
Connecting to github-production-release-asset-2e65be.s3.amazonaws.com (github-pr
oduction-release-asset-2e65be.s3.amazonaws.com)|52.216.1.232|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 150869608 (144M) [application/octet-stream]
Saving to: 'arachni-1.5.1-0.5.12-linux-x86_64.tar.gz'

arachni-1.5.1-0.5.1 100%[=====>] 143.88M  9.07MB/s   in 17s

2017-10-06 13:20:17 (8.37 MB/s) - 'arachni-1.5.1-0.5.12-linux-x86_64.tar.gz' sav
ed [150869608/150869608]

root@shortwave:~/dorkbot/tools# tar xzf arachni-1.5.1-0.5.12-linux-x86_64.tar.gz

root@shortwave:~/dorkbot/tools# ls
arachni-1.5.1-0.5.12  arachni-1.5.1-0.5.12-linux-x86_64.tar.gz  phantomjs
root@shortwave:~/dorkbot/tools# rm arachni-1.5.1-0.5.12-linux-x86_64.tar.gz
root@shortwave:~/dorkbot/tools# mv arachni-1.5.1-0.5.12/ arachni
root@shortwave:~/dorkbot/tools# cd arachni
root@shortwave:~/dorkbot/tools/arachni#
```

In the next portion of this setup, we need to create a Google [custom search engine](#). Dorkbot uses the custom search engine to locate potentially vulnerable web applications.

Don't Miss: [How to Find Vulnerable Targets Using Shodan — The World's Most Dangerous Search Engine](#)

You will need a Google account for this step. To get started, click on the "Sign in to Custom Search Engine" button. You will be prompted to enter your credentials.

The screenshot shows the Google Custom Search Engine homepage. At the top, there's a navigation bar with the Google logo, the text "Custom Search Engine", and a search box containing "Try Custom Search Engine". Below this, a heading "Make searching your site easy" is followed by a subtext: "With Google Custom Search, add a search box to your homepage to help people find what they need on your website." A "Sign in to Custom Search Engine" button is on the right. The main content area is divided into two columns. The left column has two sections: "Sign up for the basics - it's free" with a list of benefits (fast results, customization, AdSense integration) and "More control, if you need it" with a notice about the discontinuation of Google Site Search and a list of features for the paid version. The right column features a video titled "Introduction to Google Custom Search" showing a computer monitor displaying a search interface.

Make searching your site easy

With Google Custom Search, add a search box to your homepage to help people find what they need on your website.

Sign up for the basics - it's free

- Get fast and relevant search results
- Customize the look of the search results to match your site's design
- Make money off the ads we show using AdSense for Search

More control, if you need it

On April 1st, 2017, Google will discontinue the sales of Google Site Search, the paid version of Custom Search Engine. All new purchases and renewals must take place before this date. This product will be completely shut down by April 1st, 2018. This note does not affect Custom Search Engine.

- Starting at \$100 a year, increase the ways you can fine-tune the design (like opting out of the ads or the Google branding)
- Get additional powerful features for small and large businesses
- [Learn more](#)

Introduction to Google Custom Search

In order to get be able to search the entire web, we're going to have to do a bit of additional configuration on this custom search engine. First, we enter "example.com" in the *Sites to search* field. Then, we click the "Create" button to continue.

The screenshot shows the "Create CSE" page in the Google Custom Search Engine interface. The page has a sidebar on the left with links like "Edit search engine", "Help", "Help Center", "Help forum", "Support", "Blog", "Documentation", "Terms of Service", and "Send Feedback". The main content area is titled "New search engine" and contains a form. The "Sites to search" section has two input fields: the first contains "example.com" and the second contains "www.example.com". Below these fields, there's a list of suggestions: "Individual pages: www.example.com/page.html", "Entire site: www.mysite.com/*", "Parts of site: www.example.com/docs/* or www.example.com/docs/", and "Entire domain: *.example.com". A note mentions that clicking "advanced" below will allow searching for specific schema.org markups. The "Language" section has a dropdown menu set to "English". The "Name of the search engine" section has an input field containing "Example". Below this is an "Advanced Options" link. At the bottom, there's a "CREATE" button and a note: "By clicking 'Create', you agree with the Terms of Service".

New search engine

Enter the site name and click "Create" to create a search engine for your site. [Learn more](#)

Sites to search

example.com

www.example.com

You can add any of the following:

- Individual pages: `www.example.com/page.html`
- Entire site: `www.mysite.com/*`
- Parts of site: `www.example.com/docs/*` or `www.example.com/docs/`
- Entire domain: `*.example.com`

If you want to search pages over entire web containing specific schema.org markups, click on "advanced" below.

Language

English

Name of the search engine

Example

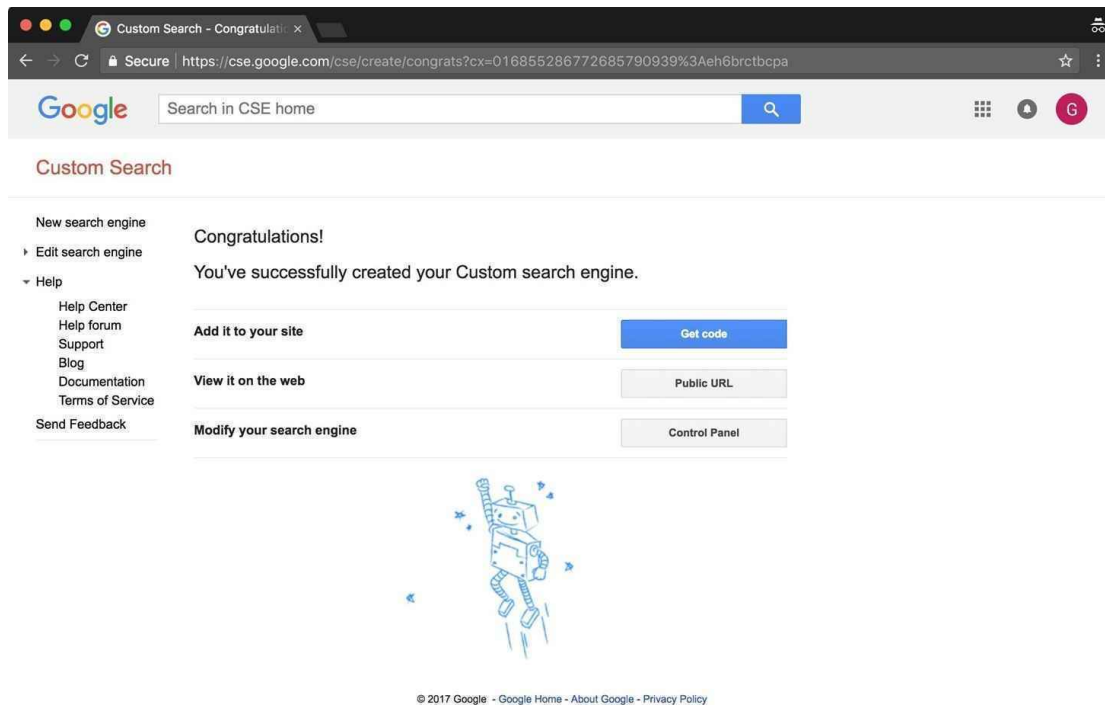
[Advanced Options](#)

By clicking 'Create', you agree with the [Terms of Service](#).

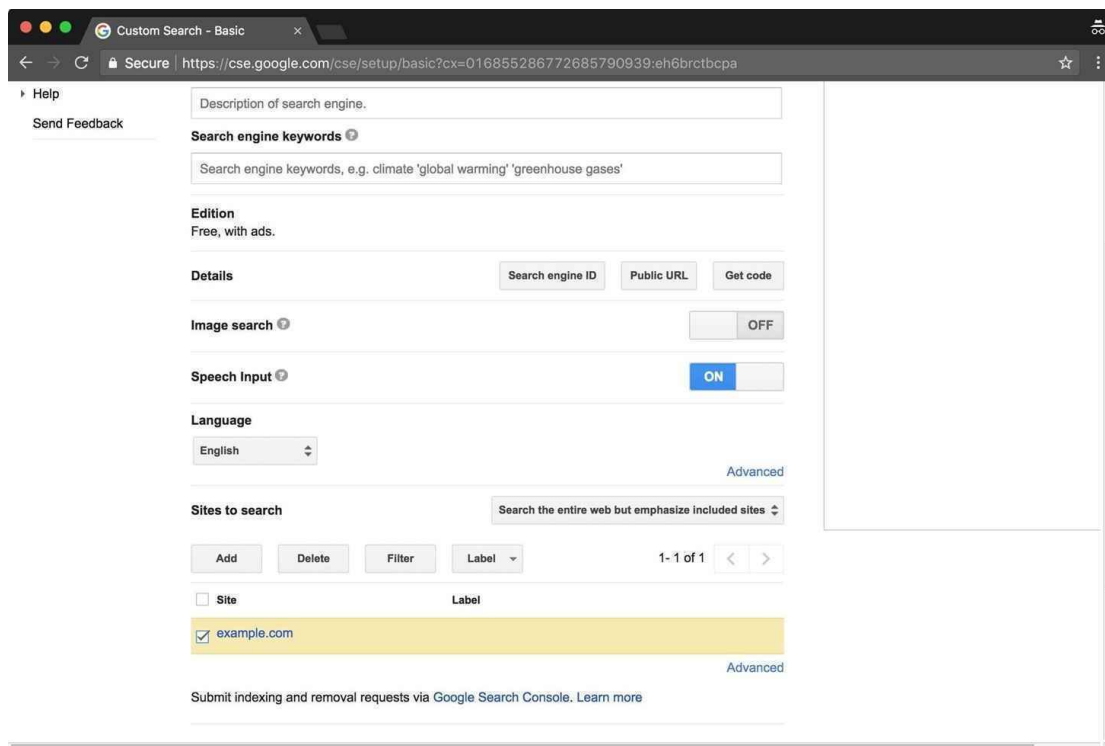
CREATE

© 2017 Google - [Google Home](#) - [About Google](#) - [Privacy Policy](#)

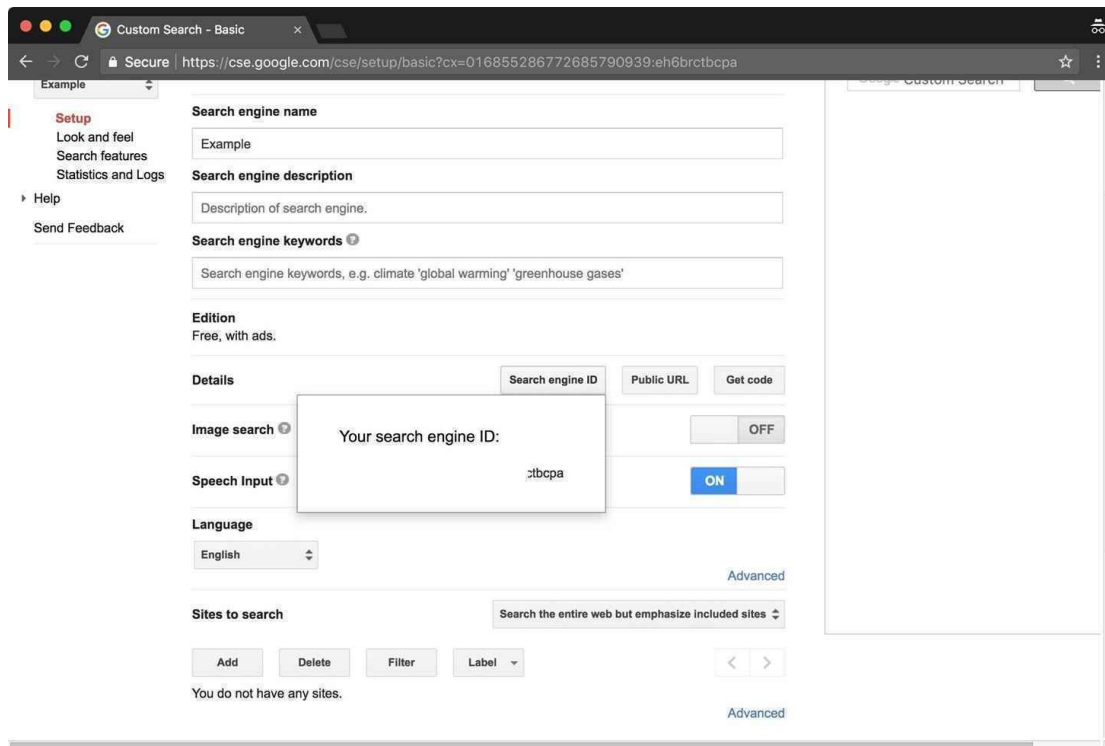
We're not done yet! This engine will only search within example.com, which isn't very useful to us. We need to change the engine to search the entire web.



Select the "Edit search engine" drop-down menu, and choose your custom search engine. Scroll down the page to the "Search only included sites" menu. Change the setting to "Search the entire web but emphasize included sites." Then, check your included site and delete it.



Lastly, we need to get the search engine ID, which we will be passing to Dorkbot. This can be found by clicking the "Search Engine ID" button.



The last step is installing Python date-util with **pip**. Do so by running the following in terminal.

```
pip install python-dateutil
```

In my case, this package was already installed.

```
1. root@shortwave: ~/dorkbot (ssh)
root@shortwave:~/dorkbot# pip install python-dateutil
Requirement already satisfied: python-dateutil in /usr/lib/python2.7/dist-packages
root@shortwave:~/dorkbot#
```

Now that we have the tool configured and installed, it's time to get down to using it.

Step 2 Run Dorkbot to Find Vulnerable Sites

Dorkbot has two distinct components: the indexer and the scanner. The indexer will search for dorks and store its findings. The scanner will follow up on those dorks and try to confirm the presence of vulnerabilities. Our first step is to scan for vulnerable sites. We'll do this by running the following in our terminal window.

```
./dorkbot.py -i google -o
engine=yourGoogleCSEHere,query="filetype:php inurl:id"
```

This will use your custom Google search engine to locate sites with PHP files and a URL containing "id."

Don't Miss: [How to Nab Free Ebooks Using Google Dorks](#)

This will not pass any results to the automated scanner. The **-i** argument tells Dorkbot to use Google as its indexer, and the **-o**

engine= is passing indexer options, telling Dorkbot to use our custom search engine. The **query=** is the query to pass to Google.

```
1. root@shortwave: ~/dorkbot (ssh)
http://www.freemoodle.org/mod/url/view.php?id=13564&redirect=1
http://atpscan.global.hornetsecurity.com/index.php?atp_q=aHR0cDovL3d3dy50cm luaXR
5Y29sbGVnZS5jb20vc2l0ZS8_aWQ9MjA5Mg&id=2092
http://us-sites.com/ID/subcat_links.php?aid=xwnadkkljcl&cat=25&sub=449&cnty=Wash
ington&ID=9593
http://www.chuhal.info/redirect.php?url=http%3A%2F%2Fdmv.org%2Fid-idaho
http://actiongames.start.bg/link.php?id=340727
https://sobesednik.ru/go.php?url=https%3A%2F%2Fwww.macrumors.com%2F2017%2F04%2F1
7%2Fiphone-8-without-touch-id-pacific-crest%2F
http://divech.start.bg/link.php?id=358126
https://moodle.ims.k12.nj.us/mod/url/view.php?id=13343
https://moodle.cdsl.qc.ca/mod/url/view.php?id=8404&redirect=1
https://www.eia.gov/tools/faqs/faq.php?id=97&t=3
http://www.puromarketing.com/actions/goexternal.php?url=http://www.gartner.com/n
ewsroom/id/3598917
https://library.med.nyu.edu/support/index.php?/Knowledgebase/Article/View/762/0/
how-do-i-activate-or-reset-the-password-for-my-net-id
http://l.pt-br.developers.prod.facebook.com/l.php?u=http%3A%2F%2Fnces.ed.gov%2Ff
astfacts%2Fdisplay.asp%3Fid%3D98&h=ATPC3I1FF0tEUbFlgVN919GUlcmJRZW54AMo-U6Xs1fnt
Fc24rlj2tzvEC9AEB0G7ccDBDy2gYw72QHrFp0cqgXryQVKwxZGhupdWTS-U1YvCP0JKwnC_KLTv1z_I
w
http://www.investorlinks.com/jump.php?id=9381
https://elearning-gilman.remote-learner.net/mod/url/view.php?id=15306
http://www.thingstodo.com/lt.php?id=10465
root@shortwave: ~/dorkbot#
```

We can use Dorkbot with the **-l** argument to list these later. So far, everything we've done has been completely acceptable. We're essentially just Google searching. We get into much trickier territory if we start using the scanner module to look for vulnerabilities. Let's try that by typing the following.

```
./dorkbot.py -i google -o engine=yourCseKeyHere,query="filetype:php inurl:id" -s arachni
```

Executing this command would pass the sites to Arachni for further processing. Depending on where you reside, executing this may be illegal. Even if it is legal, I wouldn't recommend it. Your ISP may receive an email about abuse of services, leading to a nasty phone call or potentially being dropped as a customer.

Fortunately, you can configure your Google custom search engine to search specifically within a single site that you own or have been given permission to scan. I'm going to go back and reconfigure my custom search engine to only search webscantest.com.

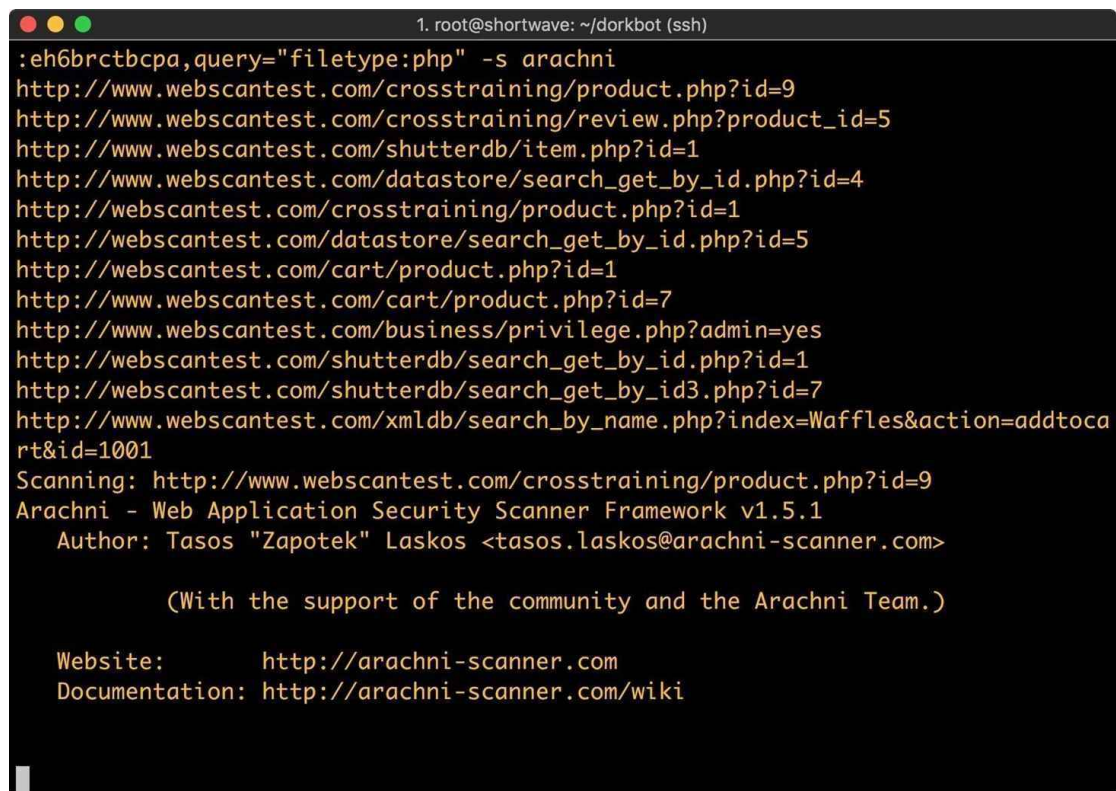
Don't Miss: How to Find Almost Every Known Vulnerability & Exploit Out There

Since Dorkbot maintains a database of returned dorks, you will need to delete this database to prevent Dorkbot from scanning hosts already in the database. We'll do so by typing these commands in terminal.

```
rm /path/to/dorkbot/databases/dorkbot.db
```

```
./dorkbot.py -i google -o
```

```
engine=yourGoogleCseHere,query="filetype:php" -s arachni
```



```
1. root@shortwave: ~/dorkbot (ssh)
:eh6brctbcpa,query="filetype:php" -s arachni
http://www.webscantest.com/crosstraining/product.php?id=9
http://www.webscantest.com/crosstraining/review.php?product_id=5
http://www.webscantest.com/shutterdb/item.php?id=1
http://www.webscantest.com/datastore/search_get_by_id.php?id=4
http://www.webscantest.com/crosstraining/product.php?id=1
http://www.webscantest.com/datastore/search_get_by_id.php?id=5
http://www.webscantest.com/cart/product.php?id=1
http://www.webscantest.com/cart/product.php?id=7
http://www.webscantest.com/business/privilege.php?admin=yes
http://www.webscantest.com/shutterdb/search_get_by_id.php?id=1
http://www.webscantest.com/shutterdb/search_get_by_id3.php?id=7
http://www.webscantest.com/xmldb/search_by_name.php?index=Waffles&action=adddtocar
rt&id=1001
Scanning: http://www.webscantest.com/crosstraining/product.php?id=9
Arachni - Web Application Security Scanner Framework v1.5.1
  Author: Tasos "Zapotek" Laskos <tasos.laskos@arachni-scanner.com>

      (With the support of the community and the Arachni Team.)

Website:      http://arachni-scanner.com
Documentation: http://arachni-scanner.com/wiki
```

Step 3 Finding Vulnerable Hosts

An attacker wishing to compromise the largest amount of systems possible in a short amount of time needs to cast a wide net. Dorkbot is designed to handle this, but how would you find vulnerabilities to target?

I recommend the [Exploit-DB](#) for this. There's an entire section dedicated to [web applications](#). For example, you could use a recently-discovered exploit in EasyBlog as your search query.

```
Release Date:
=====
2017-09-27

Product & Service Introduction:
=====
A simple and easy to setup script that allows you to have your own basic blog that comes packed with professional features.

Technical Details & Description:
=====

SQL injection on [id] parameter.

Proof of Concept (PoC):
=====

SQLi:
http://localhost/[path]/article.php?id=8' AND 7160=7160 AND 'cbgz'='cbgz

Parameter: id (GET)
Type: boolean-based blind
Title: AND boolean-based blind - WHERE or HAVING clause
Payload: id=8' AND 7160=7160 AND 'cbgz'='cbgz

=====
```

Image via Easy Blog PHP Script v1.3a

Our previous searches would include this app and many more. You will have to hone your skills with Google in order to narrow down the number of false positives. If you Google for Google dorks, you will find many more queries that you can use.

Dorks Are Useful for Mass-Scanning

If your goal is to mass-scan for vulnerabilities, Dorkbot is a solid tool worth exploring. With a bit of work on the user's part, it would be possible to almost completely automate the locating, scanning, and attacking of a particular vulnerable service. Spending some time finding semi-recent vulnerabilities and honing in on sites running specific software that is known to be exploitable could lead to many compromised machines.

Don't Miss: [How to Find Exploits Using the Exploit Database in Kali](#)

In this article, I demonstrated the configuration of a Google custom search to access the entire web. While you can do this, I wouldn't recommend it. Instead, use the custom search to scan and target domains within your control to stay legal. While working with Dorkbot, remember that it's fine to search, but connecting can be an issue.