Webinar - 6

Google Dorks

-Pre-Hacking Assessment

The contents in this pdf has proposed by "PYR RAO" in behalf of

NHC.

Note:

This content is prepared only for educational purposes, if somebody misuse it, then the team is not liable for it.

Introduction

- A Google Dork, also known as Google Dorking or Google hacking, is a valuable resource for security researchers and it has tremendous web-crawling capabilities, it can index almost anything within your website, including sensitive information
- In other words: Google “Dorking” is the practice of using Google to find vulnerable web applications and servers by using native Google search engine capabilities.

Popular Google Dork operators

Google's search engine has its own built-in query language.

The following list of queries can be run to find a list of files,

- Find information about your competition,
- Track people,
- Get information about SEO backlinks,
- Build email lists,
- Discover web vulnerabilities.

Popular Google Dorks and their work:.

- `cache`: this dork will show you the cached version of any website, e.g. `cache: securitytrails.com`
- `allintext`: searches for specific text contained on any web page, e.g. `allintext: hacking tools`
- `allintitle`: exactly the same as allintext, but will show pages that contain titles with X characters, e.g. `allintitle:"Security Companies"`
- `allinurl`: it can be used to fetch results whose URL contains all the specified characters, e.g: allinurl client area
- `filetype`: used to search for any kind of file extensions, for example, if you want to search for jpg files you can use: `filetype: jpg`
- `inurl`: this is exactly the same as `allinurl`, but it is only useful for one single keyword, e.g. `inurl: admin`
- `intitle`: used to search for various keywords inside the title, for example, `intitle:security tools` will search for titles beginning with "security" but "tools" can be somewhere else in the page.

- `inanchor`: this is useful when you need to search for an exact anchor text used on any links, e.g. `inanchor:"cyber security"`
- `intext`: useful to locate pages that contain certain characters or strings inside their text, e.g. `intext:"safe internet"`
- `link`: will show the list of web pages that have links to the specified URL, e.g. `link: microsoft.com`
- `site`: will show you the full list of all indexed URLs for the specified domain and subdomain, e.g. `site:securitytrails.com`
- `*`: wildcard used to search pages that contain "anything" before your word, e.g. `how to * a website`, will return "how to…" design/create/hack, etc… "a website".
- `|`: this is a logical operator, e.g. `"security" "tips"` will show all the sites which contain "security" or "tips," or both words.
- `+`: used to concatenate words, useful to detect pages that use more than one specific key, e.g. `security + trails`
- `-`: minus operator is used to avoiding showing results that contain certain words, e.g. `security -trails` will show pages that use "security" in their text, but not those that have the word "trails."

If you're looking for the complete set of Google operators, you can follow this SEJ post which covers almost every known dork available today.

Google Dork examples

Let's take a look at some practical examples. You'll be surprised how easy it is to extract private information from any source just by using Google hacking techniques.

Log files

Log files are the perfect example of how sensitive information can be found within any website.

Error logs, access logs and other types of application logs are often discovered inside the public HTTP space of websites.

This can help attackers find the PHP version you're running, as well as the critical system path of your CMS or frameworks.

For this kind of dork we can combine two Google operators, allintext and filetype, for example:

```
allintext:username filetype:log
```

This will show a lot of results that include username inside all *.log files.

Vulnerable web servers

The following Google Dork can be used to detect vulnerable or hacked servers that allow appending "/proc/self/cwd/" directly to the URL of your website.

```
inurl:/proc/self/cwd
```

Open FTP servers

Google does not only index HTTP-based servers, it also indexes open FTP servers.

With the following dork, you'll be able to explore public FTP servers, which can often reveal interesting things.

```
intitle:"index of" inurl:ftp
```

ENV files

.env files are the ones used by popular web development frameworks to declare general variables and configurations for local and online dev environments.

One of the recommended practices is to move these .env files to somewhere that isn't publicly accessible.

SSH private keys

SSH private keys are used to decrypt information that is exchanged in the SSH protocol. As a general security rule, private keys must always remain on the system being used to access the remote SSH server, and shouldn't be shared with anyone.

With the following dork, you'll be able to find SSH private keys that were indexed by uncle Google.

intitle:index.of id_rsa -id_rsa.pub

Email lists

It's pretty easy to find email lists using Google Dorks.

```
filetype:xls inurl:"email.xls"
```

Live cameras

Have you ever wondered if your private live camera could be watched not only by you but also by anyone on the Internet?

The following Google hacking techniques can help you fetch live camera web pages that are not restricted by IP.

Here's the dork to fetch various IP based cameras:

```
inurl:top.htm inurl:currenttime
```

To find WebcamXP-based transmissions:

```
intitle:"webcamXP 5"
```

And another one for general live cameras:

```
inurl:"lvappl.htm"
```

MP3, Movie, and PDF files

Nowadays almost no one downloads music after Spotify and Apple Music appeared on the market. However, if you're one of those classic individuals who still download legal music, you can use this dork to find mp3 files:

```
intitle: index of mp3
```

The same applies to legal free media files or PDF documents you may need:

```
intitle: index of pdf intext: .mp4
```

Weather

Google hacking techniques can be used to fetch any kind of information, and that includes many different types of electronic devices connected to the Internet.

In this case, we ran a dork that lets you fetch Weather Wing device transmissions. If you're involved in meteorology stuff or merely curious, check this out:

```
intitle:"Weather Wing WS-2"
```

The output will show you several devices connected around the world, which share weather details such as wind direction, temperature, humidity and more.

Preventing Google Dorks

There are a lot of ways to avoid falling into the hands of a Google Dork.

These measures are suggested to prevent your sensitive information from being indexed by search engines.

- Protect private areas with a user and password authentication and also by using IP-based restrictions.
- Encrypt your sensitive information (user, passwords, credit cards, emails, addresses, IP addresses, phone numbers, etc).
- Run regular vulnerability scans against your site, these usually already use popular Google Dorks queries and can be pretty effective in detecting the most common ones.
- Run regular dork queries against your own website to see if you can find any important information before the bad guys do. You can find a great list of popular dorks at the Exploit DB Dorks database.

- If you find sensitive content exposed, request its removal by using Google Search Console.
- Block sensitive content by using a robots.txt file located in your root-level website directory.

Using robots.txt configurations to prevent Google Dorking

One of the best ways to prevent Google dorks is by using a robots.txt file. Let's see some practical examples.

The following configuration will deny all crawling from any directory within your website, which is pretty useful for private access websites that don't rely on publicly-indexable Internet content.

```
User-agent: *
Disallow: /
```

You can also block specific directories to be excepted from web crawling. If you have an /admin area and you need to protect it, just place this code inside:

```
User-agent: *
Disallow: /admin/
```

This will also protect all the subdirectories inside.

Restrict access to specific files:

```
User-agent: *
Disallow: /privatearea/file.htm
```

Restrict access to dynamic URLs that contain '?' symbol

```
User-agent: *
```

```
Disallow: /*?
```

To restrict access to specific file extensions you can use:

```
User-agent: *
Disallow: /*.php$/
```

In this case, all access to .php files will be denied.

Dorking Cheat-Sheet

Google Database Hacking : Google Hacking Database (GHDB) - Google Dorks, OSINT, Recon

---

It's a theoretical part practical session will be covered during online session
[Mode of training] : Online
Via Zoom

Thanks & Regards
From,

[N00B_HACK3RS_COMMUNITY], NHC

Leader : Sumit Oneness

Co-Leader : Piyush Kaushik

Community Coordinator : PYR Rao

05-08-2020