Content Discovery

What is Content Discovery?

Content discovery is a part of reconnaissance. Content can be files such as backup or configuration, videos, pictures, administration portals or application specific features.

Common Techniques Of Content Discovery

- Manual Discovery

1. Robots.txt
2. Favicon
3. Sitemap.xml
4. HTTP Headers
5. Framework Stack

- OSINT [Google Dorking]
- OSINT [Wappalyzer]
- OSINT [Wayback Machine]
- OSINT [Github]
- OSINT [S3 Bucket]

Robots.txt

Robots.txt is a directory which gives insights to search engines which page to crawl. This files gives an initial concept which locations are allowed for public access and which are forbidden.

Favicon

Favicon can be used for content discovery if it is not replaced with custom ones. Using this technique which framework is used for building the application can be understood. OWASP hosts a database for common framework icons https://wiki.owasp.org/index.php/OWASP_favicon_database.

Command for favicon and its md5 hash value:

```
curl https://xxx.com/sites/favicon/images/favicon.ico | md5sum
```

Sitemap.xml

Sitemap.xml gives insights about what files the website administrator wishes to list by the search engine. This may list some old forgotten functionality that can lead to sensitive information leaks.

HTTP Headers

HTTP headers can sometimes leak underlying software versions and which headers are allowed.

```
curl https://x.x.x.x -v
```

Framework Stack

Poking around the applications source codes may leak the underlying framework stacks. For understanding the internal routes and architecture framework documentation can be referred to.

OSINT [Google Dorking]

Google dorking can be used to pick custom contents from an application. Below are the common filters of google dorking:

site: returns results from a specific website.
inurl: returns results that have the specific word in the url.
fietype: return results that have specific file extensions.
intitle: returns results that contains specific words in it's titles.

Reference:
[1].https://www.exploit-db.com/google-hacking-database
[2].https://www.youtube.com/watch?v=fo1BR9itwOY

OSINT [Wappalyzer]

Wappalyer is a tool which also gives results regarding the tech stack used for building the web application. But it also generates false positives so blindly refer to this can lead to rabbit holes.

Reference:
[1]. https://www.wappalyzer.com/

OSINT [Wayback Machine]

Wayback machine gives results from its large archive database. Application url can be searched from wayback machines and it will show the scraped contents. Wayback urls and gau are the two most popular tools for this purpose:

Waybackurls

```
cat domains.txt | waybackurls > urls
```

Gau

```
cat domains.txt | gau --threads 5
```

Reference:
[1]. https://github.com/tomnomnom/waybackurls
[2]. https://github.com/lc/gau

OSINT [Github]

Large teams uses github for team collaboration or version control system purpose. Often developers leaks api keys or javascript secrets or credentials exposed in the github. Github dorking can come in handy for discovering juicy stuff. Gitrob and trufflehogs are two most popular tools for this purpose.

Trufflehog

```
$ trufflehog git https://github.com/trufflesecurity/trufflehog.git
```

Gitrob

```
gitrob [options] target [target2] ... [targetN]
```

Reference:
[1]. https://www.youtube.com/watch?v=l0YsEk_59fQ

OSINT [S3 Bucket]

S3(Simple Storage Service) is a service provided by the AWS(Amazon Web Service). S3 can be used for hosting files. S3 follows this url structure:

```
http(s)://{name}.s3.amazonaws.com
```

S3 bucket can be found in the web page source, github repositories or google dorking.

Lazy S3 is popular tool for finding and exploiting misconfigured S3 buckets.

Reference:
[1].https://github.com/nahamsec/lazys3/