Raccoon

Overview

The main goal of this tool is to reconnaissance and gather information about websites. It has a wide variety of utilities, listed below:

- DNS details
- DNS visual mapping using DNS dumpster
- WHOIS information
- TLS Data - supported ciphers, TLS versions, certificate details and SANs
- Port Scan
- Services and scripts scan
- URL fuzzing and dir/file detection
- Subdomain enumeration - uses Google dorking, DNS dumpster queries, SAN discovery and bruteforce
- Web application data retrieval
- Detects known WAFs
- Supports anonymous routing through Tor/Proxies
- Uses asyncio for improved performance
- Saves output to files - separates targets by folders and modules by files

The objective of this research was to test Raccoon against several domains and analyze interesting outputs.

Tools and services used

- **Ubuntu:** An open source Debian-based Linux distribution.
- **Raccoon:** Offensive Security Tool for Reconnaissance and Information Gathering

Raccoon was downloaded using packet management tool ***pip3***

```
bestwina@bestwinaPC:~$ pip3 install raccoon-scanner
```

Tests

Port Scan

Raccoon uses nmap to run port scanning, it can use -sc and -sv flags to scan for details about services and scripts running in the found ports.

“Port scanning is one of the most popular forms of reconnaissance ahead of a hack, **helping attackers determine which ports are most susceptible**. Port scanning can lead to a hacker entering your network or stealing proprietary data” - Datto

Results:
Running instagram.com

```
[v] Nmap discovered the following ports:
  80/tcp open http
  443/tcp open https
  843/tcp closed unknown
  5222/tcp closed xmpp-client
```

It could be found two ports besides 80 and 443 that are standard http and https ports, which could lead to a possible security vulnerability

Running google.com

```
[v] Nmap discovered the following ports:
        80/tcp open http
        443/tcp open https
```

Nothing was found apart from the expected ports

WAF (Web application Firewall) Scan:

WAF is an application firewall for HTTP applications. It prevents common attacks such as SQL injection and XSS. If the WAF used is identified, we can have access to the rules that are being applied to restrict the attacks, which helps to guide attacks in the opposite direction or even to explore known vulnerabilities of the WAF used.

Results:

Running pbs.org

```
[#] Trying to detect WAF presence in pbs.org
[x] Detected WAF presence in web application: CloudFront
```

CloudFront WAF was detected, we can find general information about it on amazon site's:
https://docs.aws.amazon.com/waf/latest/developerguide/cloudfront-features.html

Running instagram.com

```
[#] Trying to detect WAF presence in www.instagram.com
[v] Did not detect WAF presence in target
```

No WAF was detected

DNS Details:

"Internal DNS servers hold all the server names an IP addresses for their domains and will share them with anyone that asks. This makes DNS a great source of information for attackers when they're trying to do internal reconnaissance."-JEFF PETTERS

Running google.com

```
142.250.68.142
20 alt1.aspmx.l.google.com.
40 alt3.aspmx.l.google.com.
10 aspmx.l.google.com.
50 alt4.aspmx.l.google.com.
30 alt2.aspmx.l.google.com.
ns3.google.com.
ns4.google.com.
ns2.google.com.
ns1.google.com.
ns1.google.com. dns-admin.google.com. 393732289 900 900 1800 60
"v=spf1 include:_spf.google.com ~all"
"facebook-domain-verification=22rm551cu4k0ab0bxsw536tlds4h95"
"docusign=1b0a6754-49b1-4db5-8540-d2c12664b289"
"apple-domain-verification=30afIBcvSuDV2PLX"
"globalsign-smime-dv=CDYX+XFHUw2wml6/Gb8+59BsH31KzUr6c1l2BPvqKX8="
"google-site-verification=wD8N7i1JTNTkezJ49swvWW48f8_9xveREV4oB-0Hf5o"
"MS=E4A68B9AB2BB9670BCE15412F62916164C0B20BB"
"google-site-verification=TV9-DBe4R80X4v0M4U_bd_J9cpOJM0nikft0jAgjmsQ"
"docusign=05958488-4752-4ef2-95eb-aa7ba8a3bd0e"
```

Running instagram.com

```
54.84.30.180
31.13.93.174
52.203.67.150
34.193.95.228
35.174.77.193
10 mxb-00082601.gslb.pphosted.com.
10 mxa-00082601.gslb.pphosted.com.
ns-868.awsdns-44.net.
ns-2016.awsdns-60.co.uk.
ns-384.awsdns-48.com.
ns-1349.awsdns-40.org.

z-p42-instagram.c10r.instagram.com.
ns-384.awsdns-48.com. awsdns-hostmaster.amazon.com. 3 7200 900 1209600 3600
"v=spf1 include:facebookmail.com include:spf.thefacebook.com include:spf-00082601.pphosted.com -all"
"adobe-idp-site-verification=367fda82-a8bb-46cf-9cff-0062d452d229"
"hyWdekepiNsp/V9b1JCR+wZDdzbESurl4GqY+FLMfiN+7aeFaway0Art+kNDHeL5OnGZipNeV/iIC+lOONSQVQ=="
"nEXgIFIbDifAKlSMQvAhly5SA-vpsAkm5wiOdwdkrzY"
"google-site-verification=GGtId51KFyq0hqX2xNvt1u0P9Xp0C7k6pp9do49fCNw"
"ms=ms86975275"
```

Web data retrieval:

Several informations can be obtained from this scan, such as:

- CMS detection
- Web server info and X-Powered-By
- robots.txt and sitemap extraction
- Cookie inspection
- Extracts all fuzzable URLs
- Discovers HTML forms

All those items can eventually provide sensitive information that could lead to a vulnerability

CMS detection

A CMS (Content Management System) is a platform which helps in creating and delivering the web applications quickly. Some CMSs are very popular and those are Wordpress, Drupal, Joomla, and vBulletin.

There are multiple known vulnerabilities in these CMSs, which weakens all websites that use the CMS, thus, knowing the CMS being used by a website, could lead to a possible exploitation, in case the version of the CMS has a known vulnerability. CMSs such as Wordpress and Joomla have already had lot of exploitations among all their versions, which can be seeing in the following links:
https://www.cvedetails.com/vulnerability-list/vendor_id-2337/product_id-4096/
https://www.cvedetails.com/vulnerability-list/vendor_id-3496/Joomla.html

Robot.txt and Sitemap.xml extraction:

Both files provide important directories of a website, searching through them could lead to sensitive information or even to special directories that could contain vulnerabilities.

Robots.txt of google.com

```
bestwina@bestwinaPC:~/Raccoon_scan_results/google.com$ strings robots.txt
User-agent: *
Disallow: /search
Allow: /search/about
Allow: /search/static
Allow: /search/howsearchworks
Disallow: /sdch
Disallow: /groups
Disallow: /index.html?
Disallow: /?
Allow: /?hl=
Disallow: /?hl=*&
Allow: /?hl=*&gws_rd=ssl$
Disallow: /?hl=*&*&gws_rd=ssl
Allow: /?gws_rd=ssl$
```

Sitemap.xml of google.com

```
bestwina@bestwinaPC:~/Raccoon_scan_results/google.com$ strings sitemap.xml
<?xml version="1.0" encoding="UTF-8"?>
<sitemapindex xmlns="http://www.google.com/schemas/sitemap/0.84">
  <sitemap>
    <loc>https://www.google.com/gmail/sitemap.xml</loc>
  </sitemap>
  <sitemap>
    <loc>https://www.google.com/forms/sitemaps.xml</loc>
  </sitemap>
  <sitemap>
    <loc>https://www.google.com/slides/sitemaps.xml</loc>
  </sitemap>
  <sitemap>
    <loc>https://www.google.com/sheets/sitemaps.xml</loc>
  </sitemap>
  <sitemap>
    <loc>https://www.google.com/drive/sitemap.xml</loc>
  </sitemap>
  <sitemap>
    <loc>https://www.google.com/docs/sitemaps.xml</loc>
  </sitemap>
  <sitemap>
    <loc>https://www.google.com/get/sitemap.xml</loc>
  </sitemap>
  <sitemap>
    <loc>https://www.google.com/flights/sitemap.xml</loc>
  </sitemap>
```

Cookie Inspection:
Exploitation of cookies is a well known vulnerability, it is used to hijack sessions, gaining unauthorized access to information or services in a computer system.

Running instagram.com

```
[v] Cookie: {csrftoken} - HttpOnly flag not set
[v] Cookie: {ig_nrcb} - HttpOnly flag not set
[v] Cookie: {mid} - HttpOnly flag not set
```

In this case, HttpOnly flag was not set, which means that the cookies can be accessed client side, opening doors to attacks such as XSS and CSRF that can hijack those cookies.

Extracts all fuzzable URLs and :

Sometimes there exists urls that are hidden and the owner doesn't want to be public. Although, fuzzing URLs makes possible to find those directories, which could lead to sensitive directories. The same is applicable to subdomains.

```
[#] Fuzzing URLs
[#] Reading from list: /home/bestwina/.local/lib/python3.8/site-packages/raccoon_src/wordlists/fuzzlist
        [405] https://www.google.com/logos
        [200] https://www.google.com/business
        [200] https://www.google.com/books
        [405] https://www.google.com/publications
        [200] https://www.google.com/crossdomain.xml
        [200] https://www.google.com/blog/wp-login.php
        [200] https://www.google.com/store
        [200] https://www.google.com/de
        [200] https://www.google.com/it
        [200] https://www.google.com/sites
        [200] https://www.google.com/news
        [200] https://www.google.com/2006
        [200] https://www.google.com/developer
        [200] https://www.google.com/settings/
        [200] https://www.google.com/search
        [200] https://www.google.com/2010
        [200] https://www.google.com/2011
        [200] https://www.google.com/blog/wp-login
```

URLs found by fuzzing in google.com

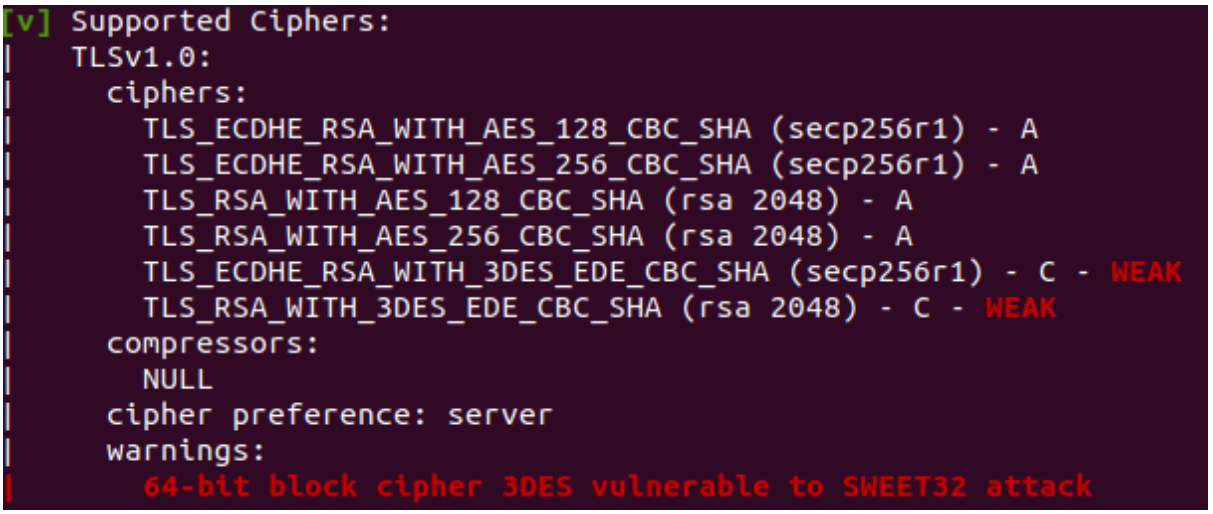
```
[!] Bruteforcing subdomains
[#] Reading from list: /home/bestwina/.local/lib/python3.8/site-packages/raccoon_src/wordlists/subdomains
        [200] https://labs.google.com
        [200] https://image.google.com
        [200] https://business.google.com
        [200] https://desktop.google.com
        [200] https://lp.google.com
        [200] https://store.google.com
        [200] https://forms.google.com
        [200] https://books.google.com
        [200] https://donate.google.com
        [200] https://sites.google.com
        [200] https://map.google.com
        [200] https://news.google.com
        [200] https://music.google.com
        [200] https://alerts.google.com
        [200] https://search.google.com
        [200] https://fi.google.com
        [200] https://fotos.google.com
```

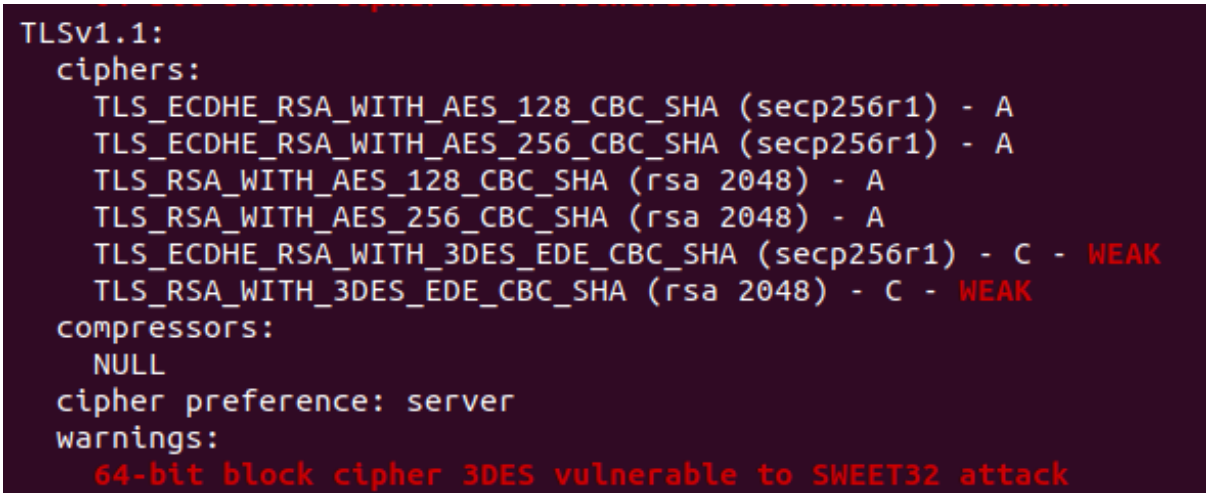
Subdomains found by fuzzing in google.com

TLS Data - supported ciphers, TLS versions, certificate details and SANs

“TLS vulnerabilities are a dime a dozen—at least so long as obsolete versions of the protocol are still in active deployment. Some major attack vectors arise from conceptual flaws in the TLS standard itself. Features prone to vulnerabilities include protocol downgrades, connection renegotiation, and session resumption” - Cloudinsidr.

Running instagram.com





Conclusion

Overall it was possible to learn all features Raccoon can provide, in addition to studying about all vulnerabilities Raccoon intended to find with it’s reconnaissance and information gathering.