

01 - Chapter Introduction



Web Application Enumeration, Revisited

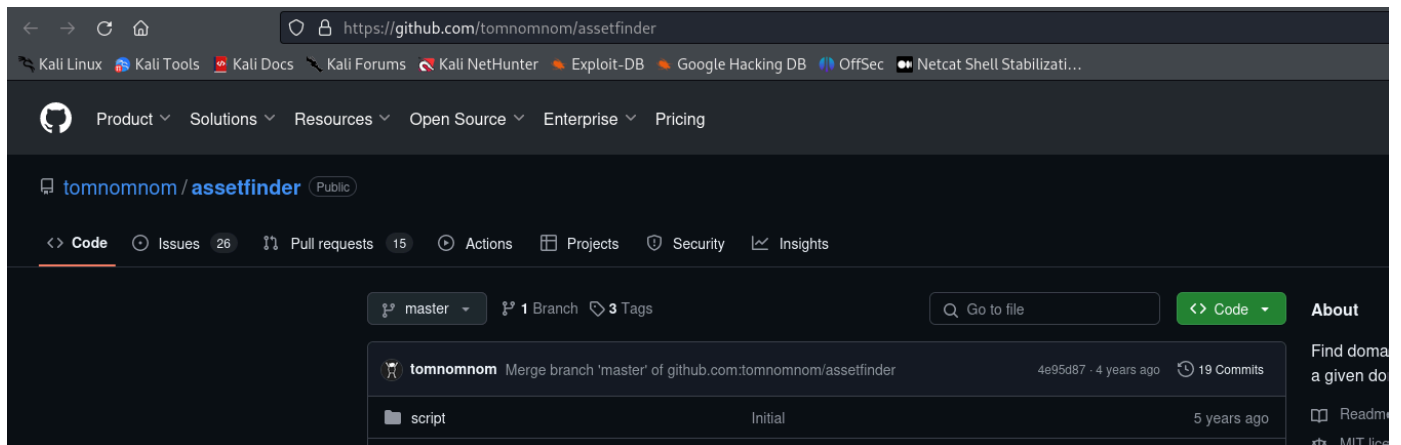
Introduction

We already talked about many of the tools we can use to enumerate websites. Here, we are going to learn how to use other ones, techniques for enumerating, and how to automate all this process. Heath is also going to provide his script for his automated process. Lets go.

Installing Go:

If we installed pimpmypkali, we should be good to go. If not, go clone the repo, and then we can run option 3. (Install go or update fix bugs...)

02 - Assetfinder - Finding Subdomains



<https://github.com/tomnomnom/assetfinder>

Stay tuned with this account.

The repo gives the command to install the tool.

"#go get -u github.com/tomnomnom/assetfinder" this does not work anymore. IT DOES NOT WORK ANYMORE

Instead use:

"#go install github.com/tomnomnom/assetfinder@latest" This one works as of 11/11/2024.

Now, we can run assetfinder command.

"#assetfinder tesla.com "

If we only want information from that subdomain, we can use:

"#assetfinder --subs-only tesla.com"

Now, we are going to create a script to run the #assetfinder command, grep the subdomain related info only, and write the output to a file:

The script:

...

#!/bin/bash

url = \$1

```
if [ ! -d "$url" ]; then
    mkdir
```

KaTeX parse error: Undefined control sequence: \[at position 11: url fi if \underline! -d "

KaTeX parse error: Undefined control sequence: \[at position 15: url fi if \underline! -d "

```
url/recon" ]; then
    mkdir $url/recon
fi
```

```
echo "[+] Harvesting subdomains with assetfinder..."
assetfinder $url >> $url/recon/assets.txt
cat $url/recon/assets.txt | grep $1 >> $url/recon/final.txt
rm $url/recon/assets.txt
```

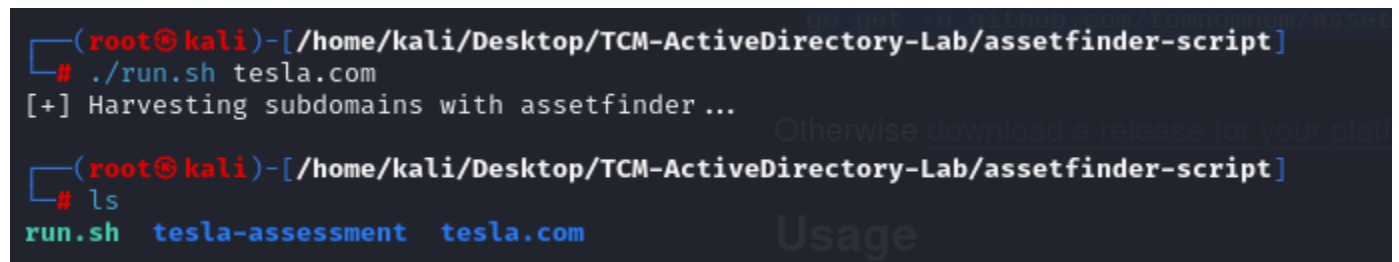
...

After generating the script, we need to run chmod:

```
"#chmod +x run.sh"
```

To run:

```
"#./run.sh tesla.com"
```



```
(root@kali)-[/home/kali/Desktop/TCM-ActiveDirectory-Lab/assetfinder-script]
# ./run.sh tesla.com
[+] Harvesting subdomains with assetfinder ...
Otherwise download a release for your platform

(root@kali)-[/home/kali/Desktop/TCM-ActiveDirectory-Lab/assetfinder-script]
# ls
run.sh  tesla-assessment  tesla.com
Usage
```

It works.

Next step, we are going to use the list generated, or the output of the assetfinder command, to run another tool and see which subdomains are alive. Which is very important information to find out.

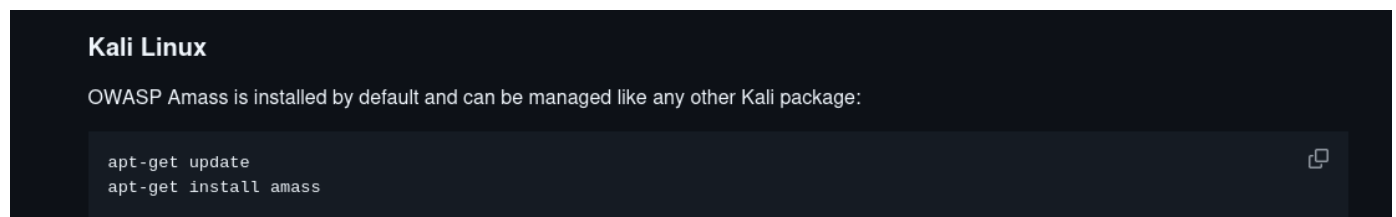
03 - Amass - Finding Subdomains

So, the idea here to learn another subdomain tool is that it uses other methods to find the subdomains for the domain name, so we could get different results with it, results that we did not pick up before with other tools.

So, we can make one script, run all the tools available, or the ones we want to run in the script, go through the list, rm the duplicate ones, and keep the rest.

<https://github.com/owasp-amass/amass>

Reading the repo:



```
Kali Linux
OWASP Amass is installed by default and can be managed like any other Kali package:
apt-get update
apt-get install amass
```

We do not need to install.

To run:

```
"#amass enum -d tesla.com"
```

So, the script we had, we are going to be incrementing on the same code. We want to have a single file with no duplicates, and we are going to use that file to probe the subdomains to see if they are alive using http

04 - Httpprobe - Finding Alive Domains

The author of the tools is tomnomnom again.

We can search for result in github.

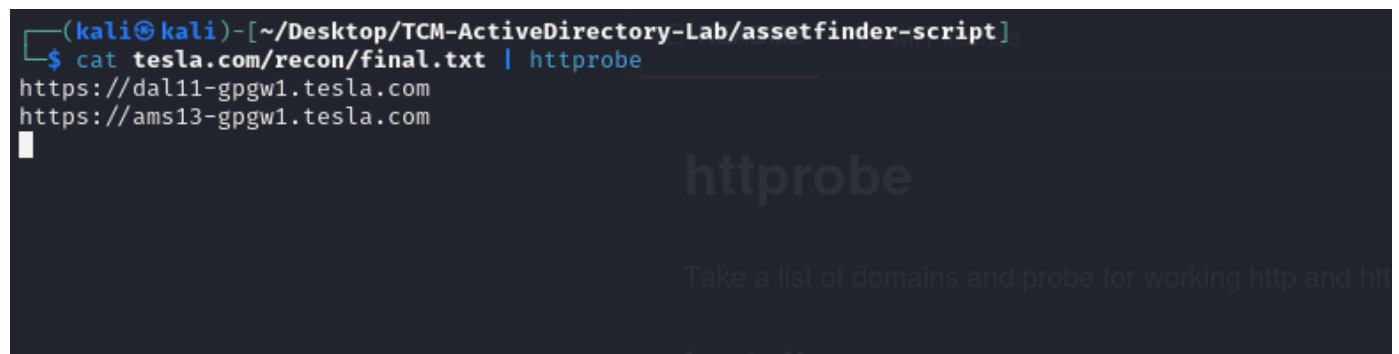
This is another Go tool.

To install we can run:

```
"#go install github.com/tomnomnom/httpprobe@latest"
```

We can just cat out the final.txt file, and pipe that to httpprobe. The domains that reply are the ones that are alive.

Just like this:

A terminal window with a dark background. The prompt is (kali@kali)-[~/Desktop/TCM-ActiveDirectory-Lab/assetfinder-script]. The command executed is \$ cat tesla.com/recon/final.txt | httpprobe. The output shows two lines: https://dal11-gpgw1.tesla.com and https://ams13-gpgw1.tesla.com. In the background, there is a large, semi-transparent watermark that says 'httprobe' and a smaller line of text below it: 'Take a list of domains and probe for working http and https'.

-s removes all the default ports.

-p listen on 443

We are going to be making both http, and https requests.

We can run:

```
"#
```

```
cat tesla.com/recon/final.txt | httpprobe -s -p https:443 | sed 's/https\?:\\V\\/' | tr -d ":443"
```

This will give us only the alive domains.

```
(kali㉿kali)-[~/Desktop/TCM-ActiveDirectory-Lab/assetfinder-script]
└─$ cat tesla.com/recon/final.txt | httpprobe -s -p https:443 | sed 's/https?:\/\:\/\/' | tr -d ":443"
dal11-gpgw1.tesla.com
ams1-gpgw1.tesla.com
iad05-gpgw1.tesla.com
hnd1-gpgw1.tesla.com
feedback.tesla.com
sin05-gpgw1.tesla.com
apigateway-message-center-ownership.tesla.com
business-ui-ownership.tesla.com
logcollector-ext.tesla.com
partners.tesla.com
digitalassets.tesla.com
warehouse.tesla.com
toolbox.tesla.com
engage.tesla.com
engage.tesla.com
logcollection.tesla.com
link.tesla.com
sso.tesla.com
mfs-supplier-gfb-stg.tesla.com
secure-static-assets.tesla.com
vmanage-alerts.tesla.com
```

httpprobe accepts line-delimited domains on stdin:

```
└─$ cat recon/example/domains.txt
example.com
example.edu
example.net
└─$ cat recon/example/domains.txt | httpprobe
http://example.com
http://example.net
http://example.edu
https://example.com
https://example.edu
https://example.net
```

sed is a command that can handles text files and strings. There are many tasks we can perform with this tool.

Now, we are going to be implementing on the script

```
(kali㉿kali)-[~/Desktop/TCM-ActiveDirectory-Lab/assetfinder-script] probe for working http and https servers
└─$ sudo ./run.sh tesla.com
[sudo] password for kali:
[+] Harvesting subdomains with assetfinder... Install
[+] Probing for alive domains...

(kali㉿kali)-[~/Desktop/TCM-ActiveDirectory-Lab/assetfinder-script]
└─$ ls
run.sh tesla-assessment tesla.com
└─$ cat tesla.com/recon/alive.txt
digitalassets.tesla.com
dal11-gpgw1.tesla.com
engage.tesla.com
engage.tesla.com
feedback.tesla.com
business-ui-ownership.tesla.com
apigateway-message-center-ownership.tesla.com
ams1-gpgw1.tesla.com
iad05-gpgw1.tesla.com
link.tesla.com
hnd1-gpgw1.tesla.com
logcollector-ext.tesla.com
mfs-supplier-gfb-stg.tesla.com
logcollection.tesla.com
partners.tesla.com
sso.tesla.com
secure-static-assets.tesla.com
toolbox.tesla.com
sin05-gpgw1.tesla.com
warehouse.tesla.com
vmanage-alerts.tesla.com
```

Basic Usage

httpprobe accepts line-delimited domains on stdin:

```
└─$ cat recon/example/domains.txt
example.com
example.edu
example.net
└─$ cat recon/example/domains.txt | httpprobe
http://example.com
http://example.net
http://example.edu
https://example.com
https://example.edu
https://example.net
```

...

#!/bin/bash

url=\$1

if [! -d "\$url"]; then

mkdir

KaTeX parse error: Undefined control sequence: \[at position 15: url fi if \underline! -d "

```
url/recon" ]; then
    mkdir $url/recon
fi
```

```
echo "[+] Harvesting subdomains with assetfinder..."
assetfinder $url >> $url/recon/assets.txt
cat $url/recon/assets.txt | grep $1 >> $url/recon/final.txt
rm $url/recon/assets.txt
```

```
#echo "[+] Harvesting subdomains with Amass..."
#amass enum -d $url >> $url/recon/f.txt
#sort -u $url/recon/f.txt >> $url/recon/final.txt
#rm $url/recon/f.txt
```

```
echo "[+] Probing for alive domains..."
cat $url/recon/final.txt | sort -u | httpprobe -s -p https:443 | sed 's/https\?:\\V\\/' | tr -d ":443" >>
$url/recon/alive.txt
```

...

Now, we can go and enumerate these. We can do some greps to see if we find a something that jumps the eye.

```
view.email.tesla.com
www.tesla.com
root@kali:~# cat tesla.com/recon/alive.txt | grep dev
sso-dev.tesla.com
root@kali:~# cat tesla.com/recon/alive.txt | grep test
root@kali:~# cat tesla.com/recon/alive.txt | grep stag
root@kali:~# cat tesla.com/recon/alive.txt | grep admin
root@kali:~#
```

05 - GoWitness - Screenshotting Websites

GoWitness - <https://github.com/sensepost/gowitness>

This tool goes to a website address(subdomains and maybe even the different directories), and takes a screen shot of it. Very hand to handle big websites, so we can get a first good look at it.

We can install this tool with:

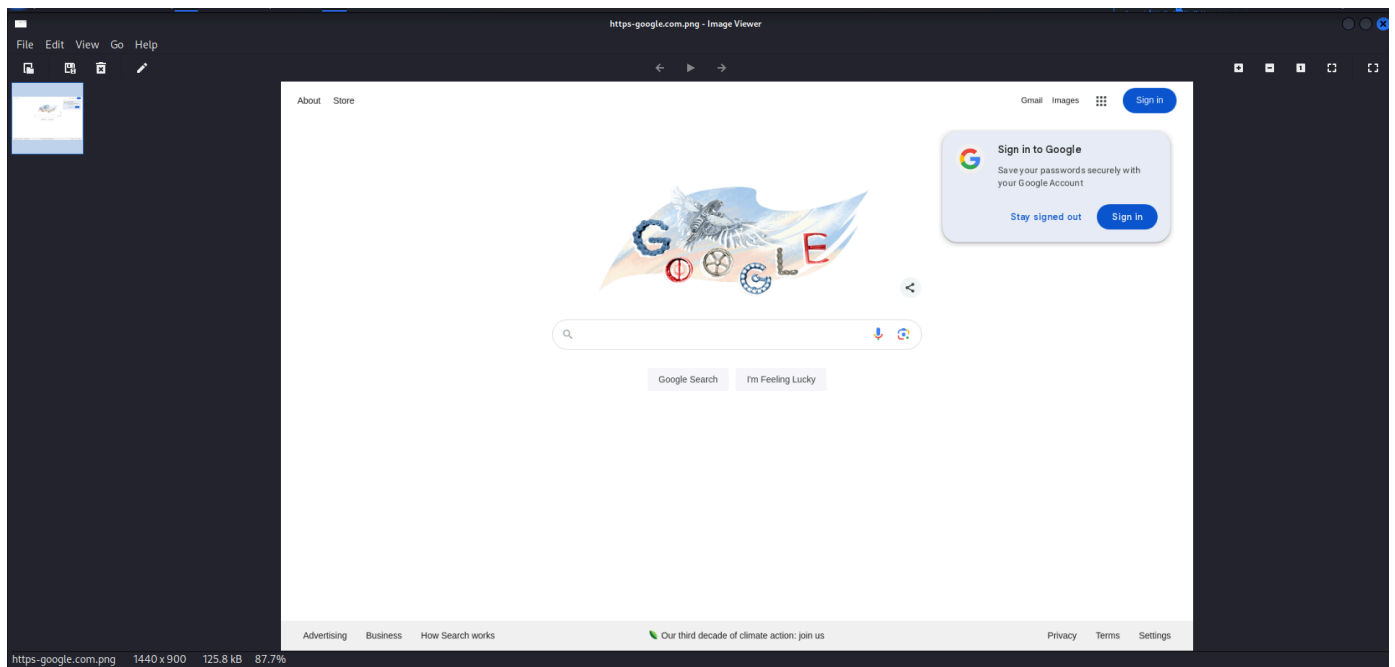
```
"#go install github.com/sensepost/gowitness@latest"
```

To run the tool we can use:

```
"#gowitness single https://tesla.com"
```

For some reason it is not working with tesla.com.

It did work with google.



When I went to the folder to open the image, I did not have privileges to access, and I am not sure how to open the file manager in sudo mode, if there is such thing. I know that I ended up opening the image with xdg-open command through the terminal.

```
"#xdg-open https-google.com.png "
```


We can run the command through a list, and automate this process as well as any other enumeration we want to perform.

Next section is going to cover this automation process.

06 - Automating the Enumeration Process

This is not only for web application, but this should be considered for all the enumeration process involved in the Pentest.

Resources for this video:

sumrecon: <https://github.com/thatonetester/sumrecon>

TCM's modified script - <https://pastebin.com/MhE6zXVt>

Subdomain takeover attack is when we actually buy the domain name, and own that subdomain. So, we would have a subdomain within their domain. We can use subjack for to check on the vulnerability.

07 - Additional Resources

The Bug Hunter's Methodology - <https://www.youtube.com/watch?v=uKWu6yhnhbQ>

Nahamsec Recon Playlist - <https://www.youtube.com/watch?v=MIujSpuDtFY&list=PLKAaMVNxvLmAkqBkzFaOxqs3L66z2n8LA>

Quiz

2 / 3

What happens following a successful subdomain takeover?

The attacker can update DNS entries

The attacker has control of an organization's entire domain

correct

The attacker has control of an organization's subdomain



◀ Back

Continue ▶

What insights can the wayback machine provide?

correct

Information such as source code, credentials or keys



Current technology stack being used

What extra security controls currently exist (e.g. WAF)

◀ Back

Continue ▶