WEB APPLICATION ENUMERATION

Wednesday, 23 June 2021 07:17

INSTALLING GO

A lot of tools we will use wil utilize the go language.
Google and search go lang and download
Download the linux version amd 64
Open terminal

```
   root@kali:~/Downloads#Cd Downloads/
   root@kali:~/Downloads#tar -xvf gol.13.
   root@kali:~/Downloads# tar -xvf gol.13.5.linux-amd64.tar.gz -C usr/local
```

Changing the owner of root directory

```
   root@kali:~/Downloads# chown -R root:root /usr/local/go
```

Making sure our paths are set correctly

```
   root@kali:~/Downloads# gedit -/.profile
```

Now a notepad will open up;

Edit your ~/.bashrc and add:

```
export GOPATH=$HOME/go
export GOROOT=/usr/local/go
export PATH=$PATH:$GOROOT/bin:$GOPATH/bin
```

Save and type “source ~/.bashrc”

To update our profile;

```
root@kali:-/Downloads# ~/.profile
```

Now lets echo the path to see if its correct

```
root@kali:-Downloads#  echo $PATH
root@kali:-Downloads# go
root@kali:-Downloads# go version
```

Finding Subdomains with Assetfinder

Type assetfinder in google and click the link to github and install it using terminal

```
root@kali:-# go get -u github.com/tomnomnom/assetfinder
root@kali:-# assetfinder tesla.com
root@kali:-# assetfinder tesla.com >> tesla-subs.txt
root@kali:-# cat tesla-subs.txt | wc -l
```

To reduce the word count use;

```
root@kali:-# assetfinder --subs-only tesla.com
```

Now write a tool that will aid us in specifying the required searches;

```
root@kali:-# gedit run.sh
#!/bin/bash
url=$1
if [ ! -d "$url" ];then
mkdir $url
fi
if [ ! -d "$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
```

To prove the tool is not there

```
root@kali:-# ls
root@kali:-# chmod +x run.sh
root@kali:-# ./run.sh tesla.com
root@kali:-# cd tesla.com/recon/
root@kali:-/tesla.com/recon# ls
root@kali:-/tesla.com/recon# gedit final.txt
```

Finding Subdomains with Amass

Amass is used for subdomain hunting.

Search for amass github on google

```
root@kali:-# export G0111MODULE=on
root@kali:-# go get -v -u github.com?OWASP/Amass/v3…
root@kali:-# amass enum -d tesla.com
```

```
#!/bin/bash
url=$1
if [ ! -d "$url" ];then
mkdir $url
fi
if [ ! -d "$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 "[+] Double checking for 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
```

Finding alive domains with Http probe

We will use http probe as our tool

Google httprobe github and search for it

```
root@kali:-# go get -u github.com/tomnomnom/httprobe
root@kali:-# cat tesla.com/recon/final.txt
root@kali:-# cat tesla.com/recon/final.txt | httprobe
root@kali:-# cat tesla.com/recon/final.txt | httprobe -s -p https:443
-s to remove default ports
root@kali:-# cat tesla.com/recon/final.txt | httprobe -s -p https:443 | sed 's/https\?:\/\///' | tr -d
':443'
```

This will pull down the https at the front and 443 leaving behind the subdomains

```
#!/bin/bash
url=$1
if [ ! -d "$url" ];then
mkdir $url
fi
if [ ! -d "$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 "[+] Double checking for 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 | httprobe -s -p https:443 | sed 's/https\?:\/\///' | tr -d ':443' >>
$url/recon/httprobe/a.txt
sort -u $url/recon/httprobe/a.txt > $url/recon/httprobe/alive.txt
rm $url/recon/httprobe/a.txt
```

```
root@kali:-# /run.sh tesla.com
root@kali:-# cat tesla.com/recon/alive.txt
root@kali:-# cat tesla.com/recon/alive.txt | grep dev
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
```

Screenshotting websites with Go witness

Go witness is another tool like eye witness.

gowitness is a website screenshot utility written in Golang, that uses Chrome Headless to generate screenshots of web interfaces using the command line, with a handy report viewer to process results.

```
root@kali:-# go get -u github.com/sensepost/gowitness
root@kali:-# gwitness --help
root@kali:-# gowitness single --url=https://tesla.com
```

A screenshot will be saved

Automating the Enumeration process

```
#!/bin/bash
url=$1
if [ ! -d "$url" ];then
mkdir $url
fi
if [ ! -d "$url/recon" ];then
mkdir $url/recon
fi
# if [ ! -d '$url/recon/eyewitness' ];then
# mkdir $url/recon/eyewitness
# fi
if [ ! -d "$url/recon/scans" ];then
mkdir $url/recon/scans
fi
if [ ! -d "$url/recon/httprobe" ];then
mkdir $url/recon/httprobe
fi
if [ ! -d "$url/recon/potential_takeovers" ];then
mkdir $url/recon/potential_takeovers
fi
if [ ! -d "$url/recon/wayback" ];then
mkdir $url/recon/wayback
fi
if [ ! -d "$url/recon/wayback/params" ];then
mkdir $url/recon/wayback/params
fi
if [ ! -d "$url/recon/wayback/extensions" ];then
mkdir $url/recon/wayback/extensions
fi
if [ ! -f "$url/recon/httprobe/alive.txt" ];then
touch $url/recon/httprobe/alive.txt
fi
if [ ! -f "$url/recon/final.txt" ];then
touch $url/recon/final.txt
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 "[+] Double checking for 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 | httprobe -s -p https:443 | sed 's/https\?:\/\///' | tr -d ':443' >>
$url/recon/httprobe/a.txt
sort -u $url/recon/httprobe/a.txt > $url/recon/httprobe/alive.txt
rm $url/recon/httprobe/a.txt

echo "[+] Checking for possible subdomain takeover..."

if [ ! -f "$url/recon/potential_takeovers/potential_takeovers.txt" ];then
touch $url/recon/potential_takeovers/potential_takeovers.txt
fi

subjack -w $url/recon/final.txt -t 100 -timeout 30 -ssl -c ~/go/src/github.com/haccer/subjack/fingerprints.json -v 3 -o
$url/recon/potential_takeovers/potential_takeovers.txt

echo "[+] Scanning for open ports..."
nmap -iL $url/recon/httprobe/alive.txt -T4 -oA $url/recon/scans/scanned.txt

echo "[+] Scraping wayback data..."
cat $url/recon/final.txt | waybackurls >> $url/recon/wayback/wayback_output.txt
sort -u $url/recon/wayback/wayback_output.txt
```

```
echo "[+] Pulling and compiling all possible params found in wayback data..."
cat $url/recon/wayback/wayback_output.txt | grep '?*=' | cut -d '=' -f 1 | sort -u >>
$url/recon/wayback/params/wayback_params.txt
for line in $(cat $url/recon/wayback/params/wayback_params.txt);do echo $line'=';done

echo "[+] Pulling and compiling js/php/aspx/jsp/json files from wayback output..."
for line in $(cat $url/recon/wayback/wayback_output.txt);do
ext="${line##*.}"
if [[ "$ext" == "js" ]]; then
echo $line >> $url/recon/wayback/extensions/js1.txt
sort -u $url/recon/wayback/extensions/js1.txt >> $url/recon/wayback/extensions/js.txt
fi
if [[ "$ext" == "html" ]];then
echo $line >> $url/recon/wayback/extensions/jsp1.txt
sort -u $url/recon/wayback/extensions/jsp1.txt >> $url/recon/wayback/extensions/jsp.txt
fi
if [[ "$ext" == "json" ]];then
echo $line >> $url/recon/wayback/extensions/json1.txt
sort -u $url/recon/wayback/extensions/json1.txt >> $url/recon/wayback/extensions/json.txt
fi
if [[ "$ext" == "php" ]];then
echo $line >> $url/recon/wayback/extensions/php1.txt
sort -u $url/recon/wayback/extensions/php1.txt >> $url/recon/wayback/extensions/php.txt
fi
if [[ "$ext" == "aspx" ]];then
echo $line >> $url/recon/wayback/extensions/aspx1.txt
sort -u $url/recon/wayback/extensions/aspx1.txt >> $url/recon/wayback/extensions/aspx.txt
fi
done

rm $url/recon/wayback/extensions/js1.txt
rm $url/recon/wayback/extensions/jsp1.txt
rm $url/recon/wayback/extensions/json1.txt
rm $url/recon/wayback/extensions/php1.txt
rm $url/recon/wayback/extensions/aspx1.txt
#echo "[+] Running eyewitness against all compiled domains..."
#python3 EyeWitness/EyeWitness.py --web -f $url/recon/httprobe/alive.txt -d $url/recon/eyewitness --resolve
```