Recon Like A Boss

More Targets- More Options- More Opportunities

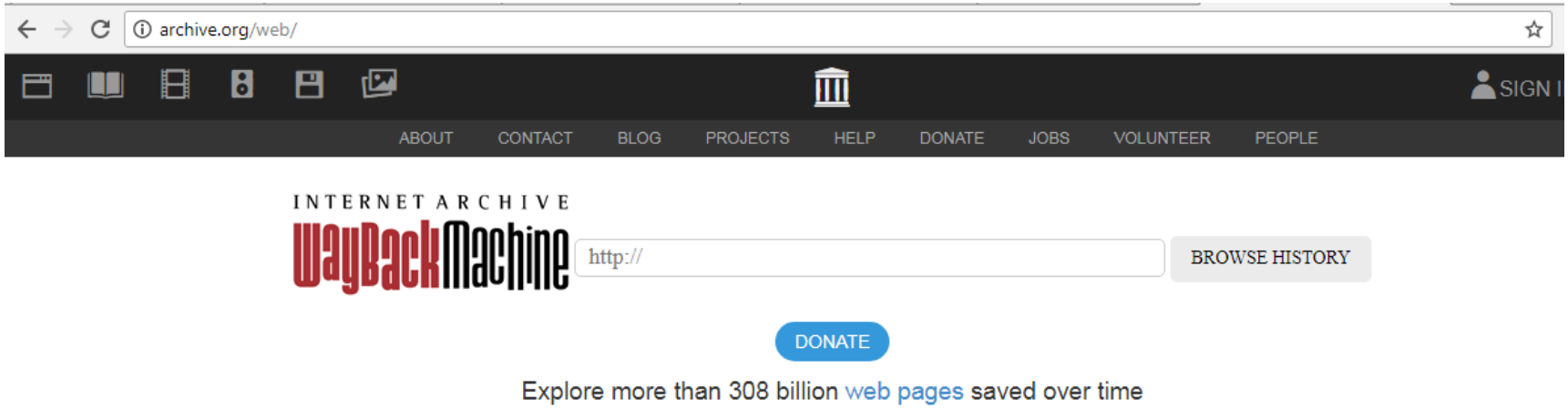
AGENDA

- Increase Your Attack Area
- Determine Technologies used by Website.
- Amazon Web Service (AWS) Recon & Hacking
- Github Recon
- Content Discovery

Increase Your Attack Area

Recon- Go Back in Time

- Wayback Machine to view old files like robots.txt and URLs



Recon- Go Back in Time

- Tools are out to automate this
- **waybackurls.py**

  Download:
  https://gist.github.com/mhmdiaa/adf6bff70142e5091792841d4b372050

- **waybackrobots.py**

  Download:

  https://gist.github.com/mhmdiaa/2742c5e147d49a804b408bfed3d32d07

Now We Have

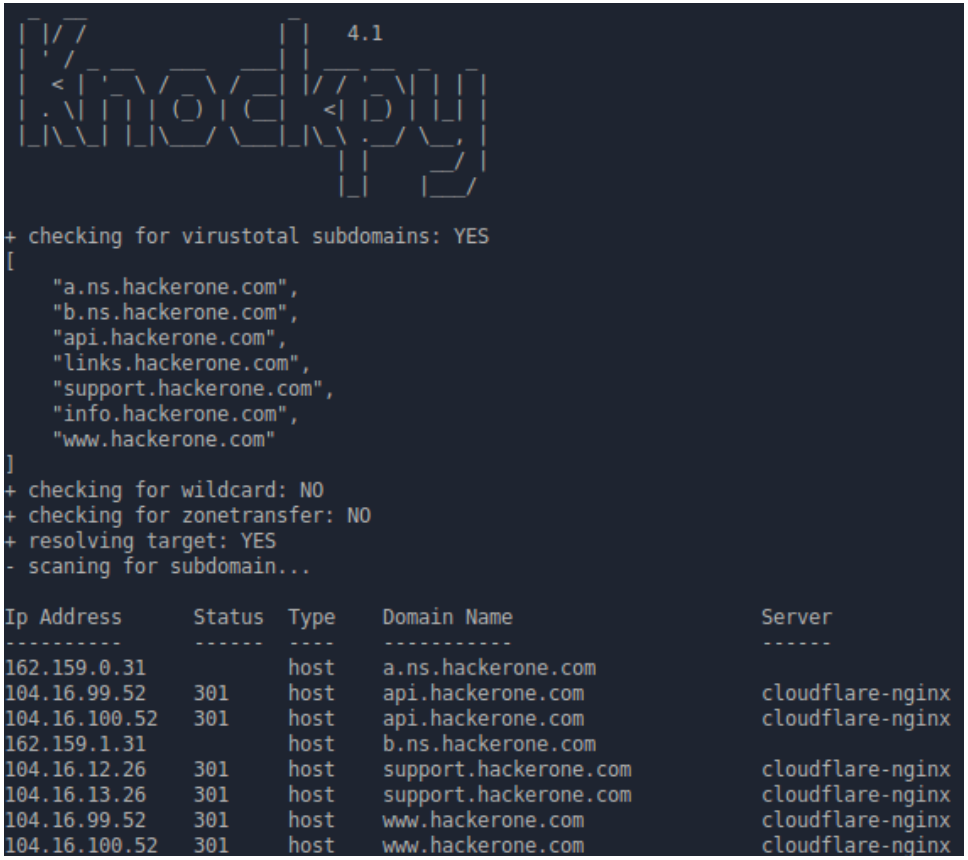
Waybackurls

Sub-domains Discovery

- Brute force on main domain
- Some scripts to automate this task
  - Knockpy:-
    https://github.com/guelfoweb/knock
  - Sublist3r:-
    https://github.com/aboul3la/Sublist3r
  - SubBrute
    https://github.com/TheRook/subbrute

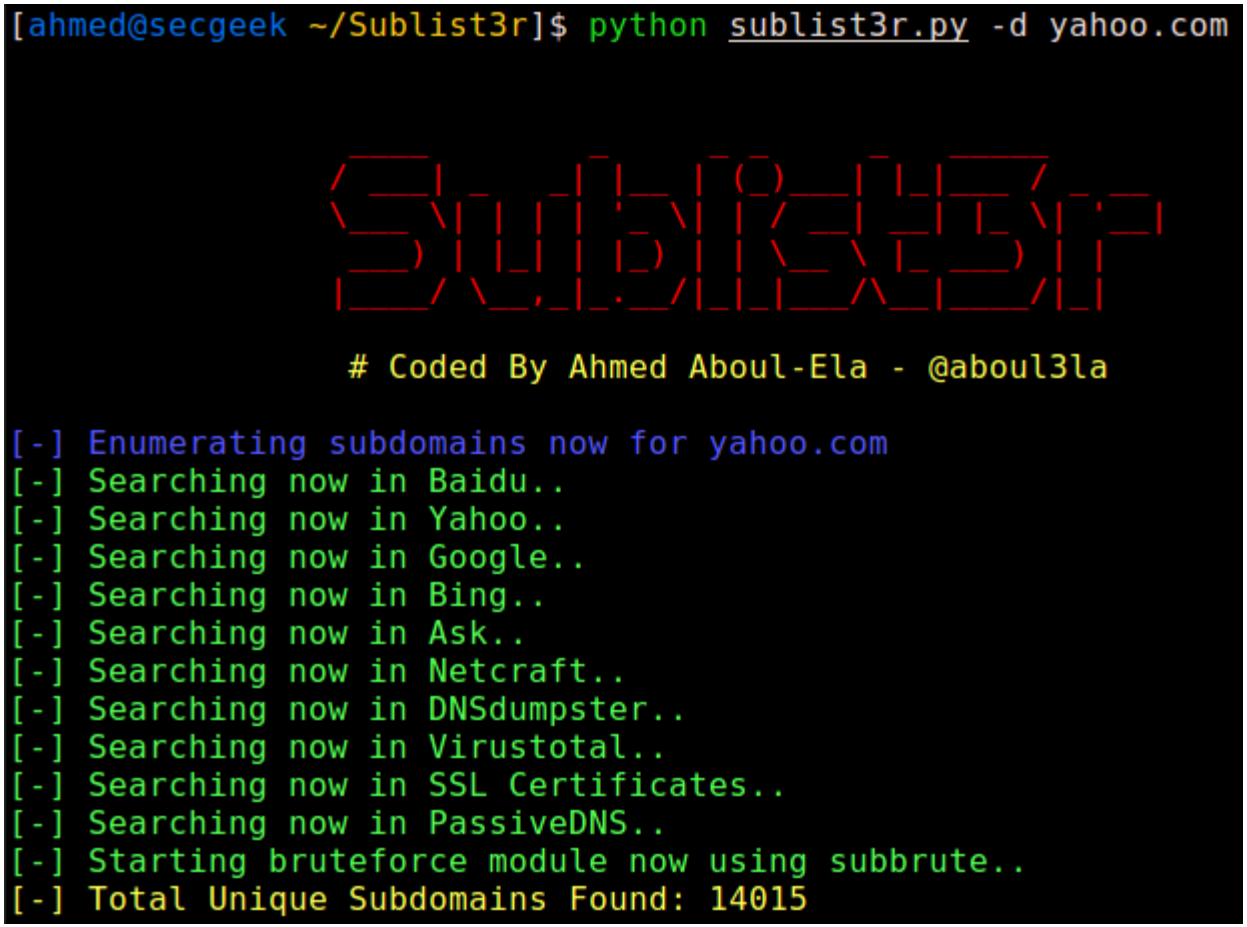
Sub-domains Discovery Knockpy

- Usage: ./knockpy target.com



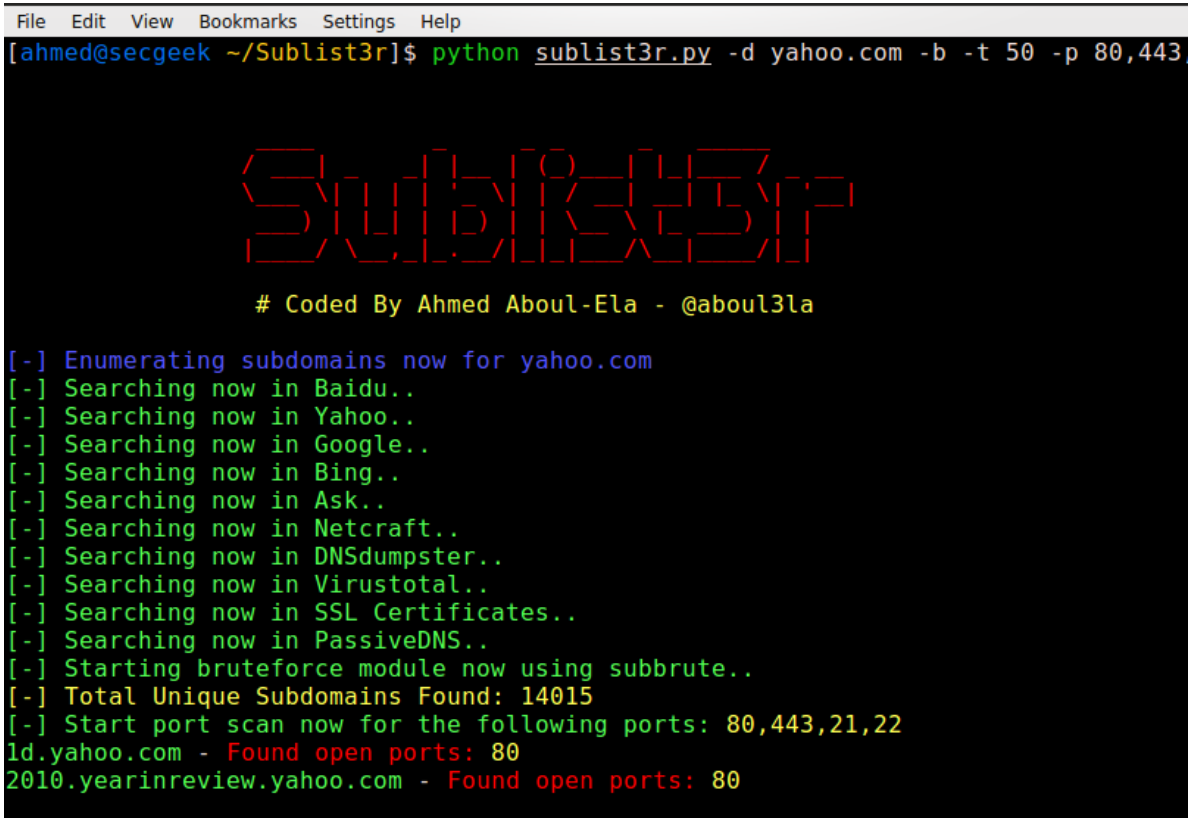
Sub-domains Discovery Sublist3r

- Usage: python sublist3r.py -d target.com



Sub-domains Discovery Sublist3r Cont.

- Find sub-domains with specific open ports
- Usage: python sublist3r.py -d target.com -p 80,443



Sub-domains Discovery SubBrute.

- Usage: ./subbrute.py google.com

- You can give list of domains like this

  Usage: ./subbrute.py -t list.txt

Sub-domains Discovery Cont.

- Google Dork

  site:target.com –site www.target.com
- Online Resource:
  - https://dnsdumpster.com/
  - https://searchdns.netcraft.com/
  - https://www.virustotal.com (Go to search and type target.com)
  - https://crt.sh/?q=%25paypal.com

    (Use “%target.com”.)

Now We Have

WaybackURls

+

Subdomains

Don’t Stop Here

Find Sub-domains of Sub-domain

http://bf1-adxdb-001.data.bf1.yahoo.com/about.php

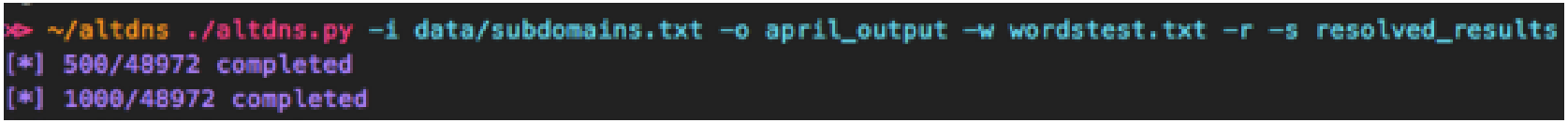
Some website have 5th and 6th level sub-domain

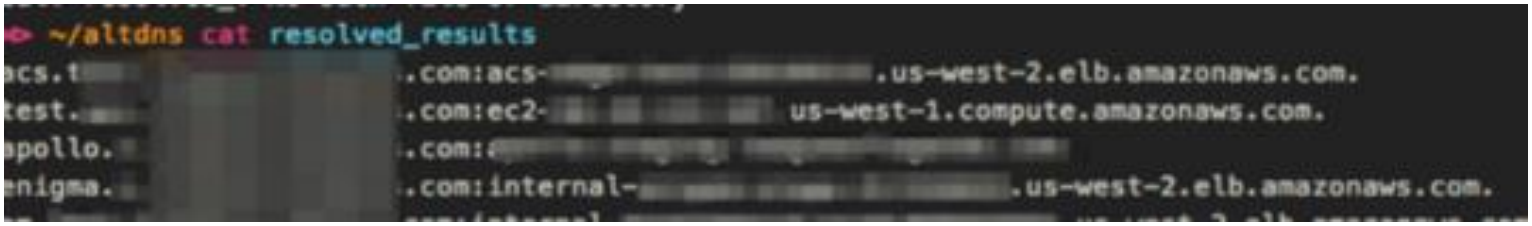
Find Subdomains of Subdomain

Tool: altdns (https://github.com/infosec-au/altdns)

Input : sub-domain list

Usage: **./altdns.py -i subdomains.txt -o data_output -w words.txt -r -s output.txt**





Find Subdomains of Subdomain

Tool: SubBrute

Usage:

./subbrute.py target.com > sudomains.txt

Then

./subbrute.py –t subdomains.txt

Now We Have

WaybackURls

+

Subdomains

+

Subdomains of Subdomains

Sub-domain Validation

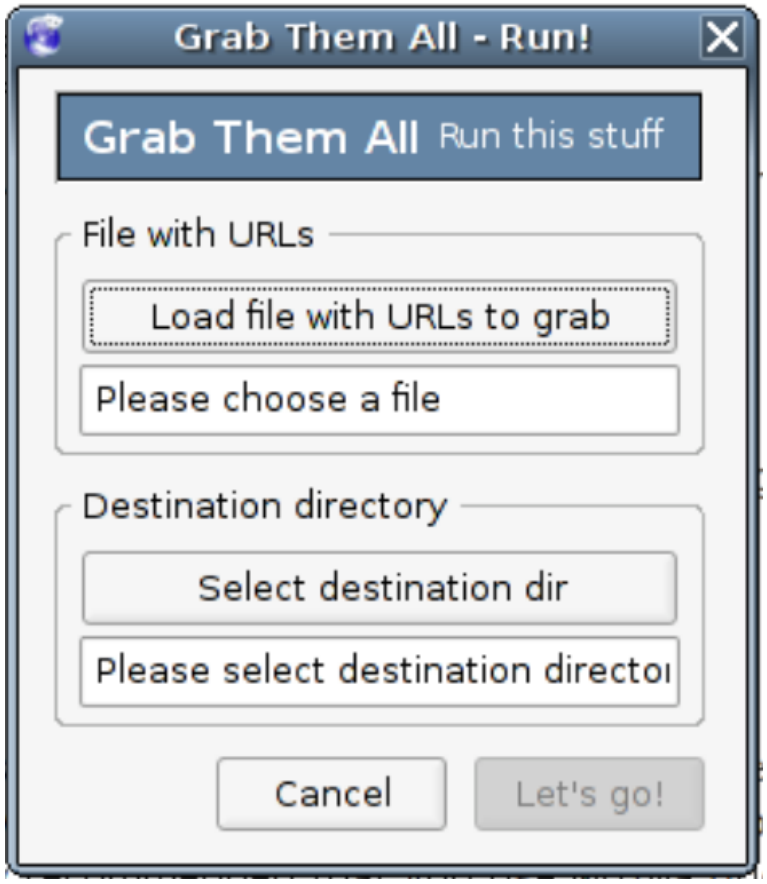
Tool: EyeWitness (https://github.com/ChrisTruncer/EyeWitness)

Provide list of sub-domains and it will give you report with screenshots of sub-domain

Usage: ./EyeWitness.py -f subdomains.txt

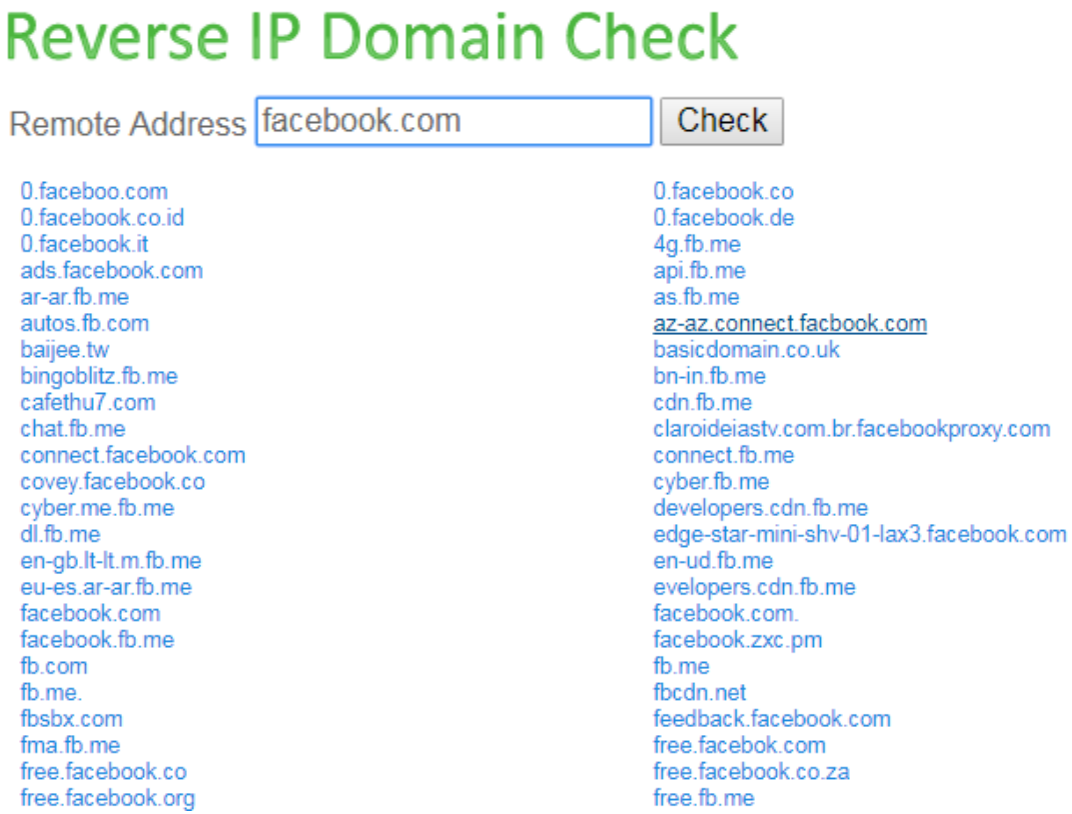
Sub-domain Validation

- Tool: Grab Them All (Mozilla addon)



Other sites on the same domain

- www.yougetsignal.com



Now We Have

WaybackURls

+

Subdomains

+

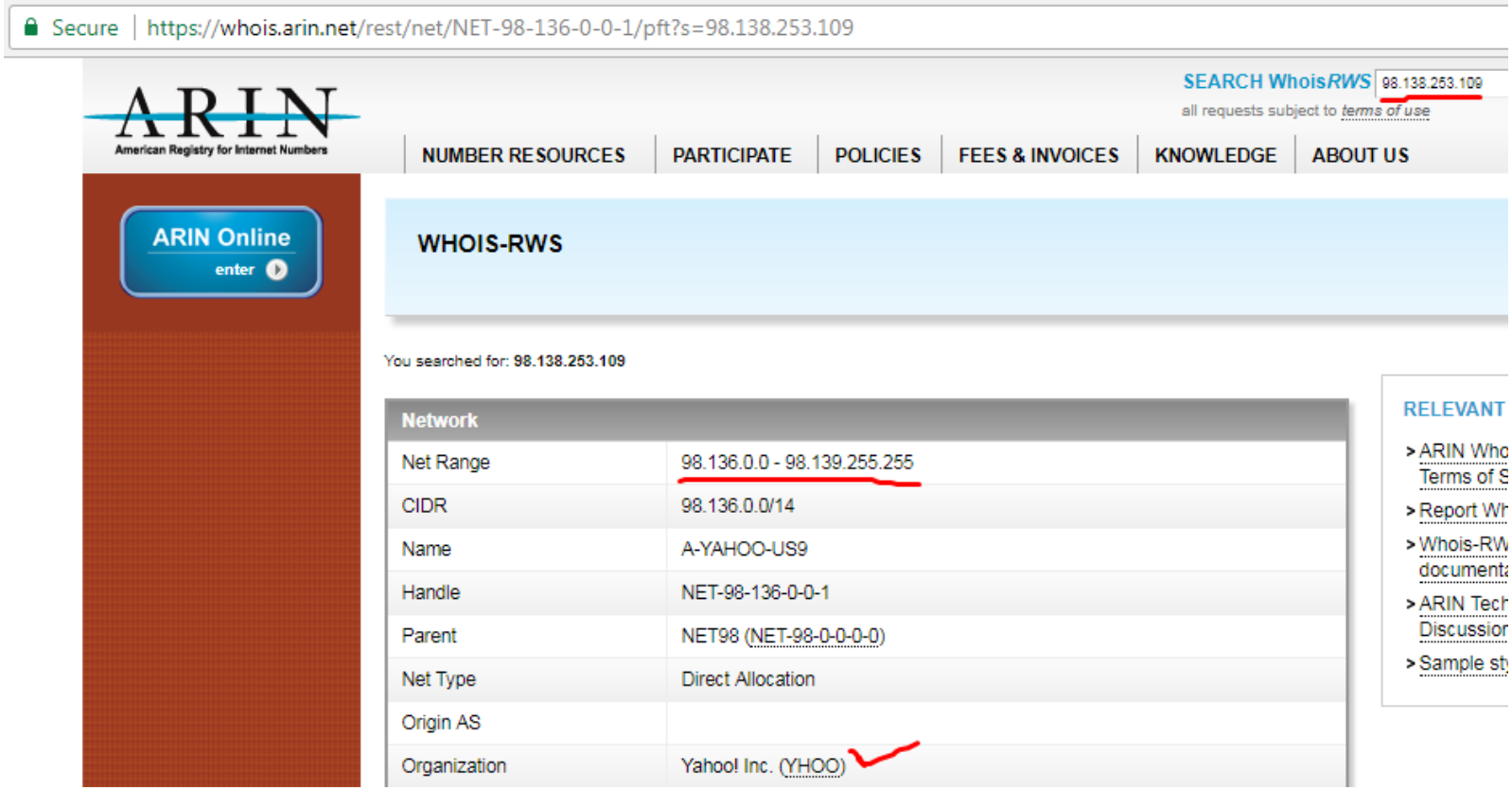
Subdomains of Subdomains

+

Other Sites on the same Domain

Target IP Range

- Url: https://whois.arin.net
- Search by Target IP



IP Range of Target Cont.

- Yahoo! owns a massive block of IP addresses
- From 98.136.0.0 - 98.139.255.255
- Which is 260,000 unique IP addresses

Got Huge IP Range

Real Case Study

- Patrik Fehrenbach (@ITSecurityguard)

  Wrote a Bash script to download **phpinfo.php** file (if found) from Yahoo! IP range

  (98.136.0.0 - 98.139.255.255)

Real Case Study

- And Finally

http://nc10.n9323.mail.ne1.yahoo.com/phpinfo.php

PHP Version 5.2.17

System	[redacted] 2.6.18-274.7.1.el5 #1 SMP Thu Oct 20 16:21:01 EDT 2011 x86_64
Build Date	Nov 8 2011 22:58:16
Configure Command	'./configure' '--enable-bcmath' '--enable-calendar' '--enable-dbase' '--enable-exif' '--enable-ftp' '--enable-gd-native-ttf' '--enable-libxml' '--enable-magic-quotes' '--enable-mbstring' '--enable-pdo=shared' '--enable-soap' '--enable-sockets' '--enable-sqlite-utf8' '--enable-zend-multibyte' '--enable-zip' '--prefix=/usr' '--with-bz2' '--with-curl=/opt/curlssl/' '--with-curlwrappers' '--with-freetype-dir=/usr' '--with-gd' '--with-gettext' '--with-imap=/opt/php_with_imap_client/' '--with-imap-ssl=/usr' '--with-jpeg-dir=/usr' '--with-kerberos' '--with-libdir=lib64' '--with-libexpat-dir=/usr' '--with-libxml-dir=/opt/xml2' '--with-libxml-dir=/opt/xml2/' '--with-mcrypt=/opt/libmcrypt/' '--with-mhash=/opt/mhash/' '--with-mime-magic' '--with-mm=/opt/mm/' '--with-mysql=/usr' '--with-mysql-sock=/var/lib/mysql/mysql.sock' '--with-mysqli=/usr/bin/mysql_config' '--with-openssl=/usr' '--with-openssl-dir=/usr' '--with-pcre-regex=/opt/pcre' '--with-pdo-mysql=shared' '--with-pdo-sqlite=shared' '--with-pic' '--with-png-dir=/usr' '--with-pspell' '--with-sqlite=shared' '--with-tidy=/opt/tidy/' '--with-ttf' '--with-xmlrpc' '--with-xpm-dir=/usr' '--with-xsl=/opt/xslt/' '--with-zlib' '--with-zlib-dir=/usr'
Server API	CGI
Virtual Directory	disabled

Bash Script

```
#!/bin/bash
for ipa in 98.13{6..9}.{0..255}.{0..255}; do
wget -t 1 -T 5 http://${ipa}/phpinfo.php; done&
```

Only 3 lines of code

Takeaways

- When hacking, consider a company's entire infrastructure. I know that Patrik has employed similar techniques to find some more.

  (Eg. Many people keep Backup.rar)

- Additionally, you'll notice there was 260,000 potential addresses here, which would have been impossible to scan manually.

- When performing this type of testing, automation is hugely important.

Now We Have

WaybackURls

+

Subdomains

+

Subdomains of Subdomains

+

Other Sites on the same Domain

+

IP Range

Find New Endpoints from JS Files

- Tools used

1. Burp Suite
2. InputScanner (Zscanner)
3. JS-Scan

Find New Endpoints from JS Files (Tools Intro.)

- Burp Suite: Proxy

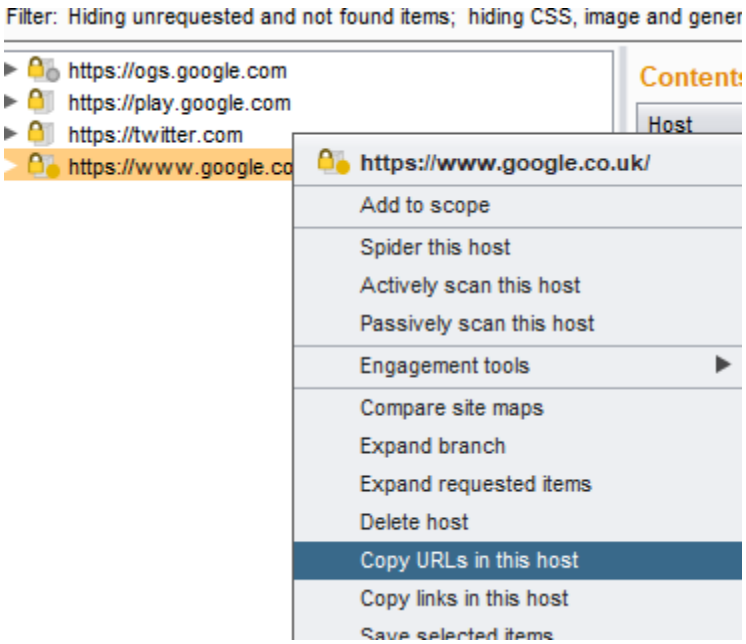
- Zscanner: A tool designed to scrape a list of URLs. This tool will also scrape .js urls found on each page

- JS-Scan: A tool designed to scrape a list of .js files and extract urls

How to use these tools together??

Find New Endpoints from JS Files (Burp Suite)

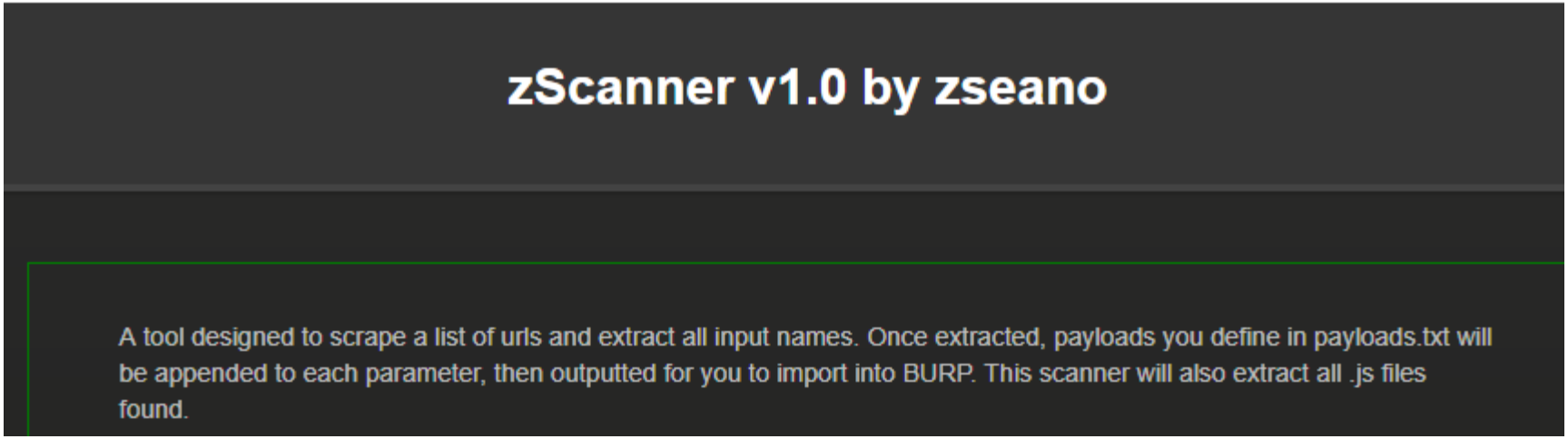
- Run Spider tool on your target in Burp Suite
- Once the spider has finished right click on the host and click "Copy Urls in this host“



Find New Endpoints from JS Files (Zscanner)

- Once copied, paste them into urls.txt
- Put urls.txt file in the root of Zscanner

  Eg. c/xampp/htdocs/zscanner/urls.txt
- Now open zscanner in browser



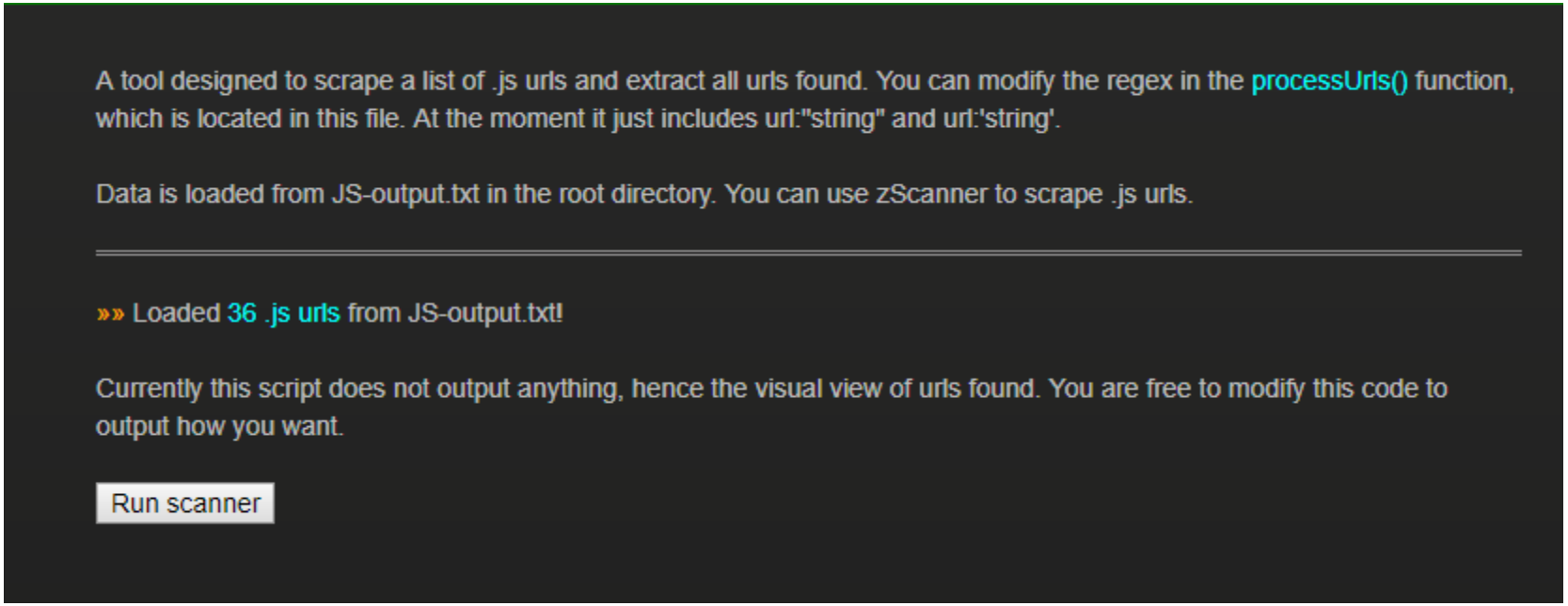
Find New Endpoints from JS Files (Zscanner.)

- Click on “Begin Scanner”
- 4 files are outputted in the /outputs/ folder: JS-output.txt, GET-output.txt, POSTHost-output.txt, POSTData-output.txt
- Copy JS-output.txt file and put it in the root of JS-Scan root folder

  Eg. c/xampp/js-scan/JS-output.txt

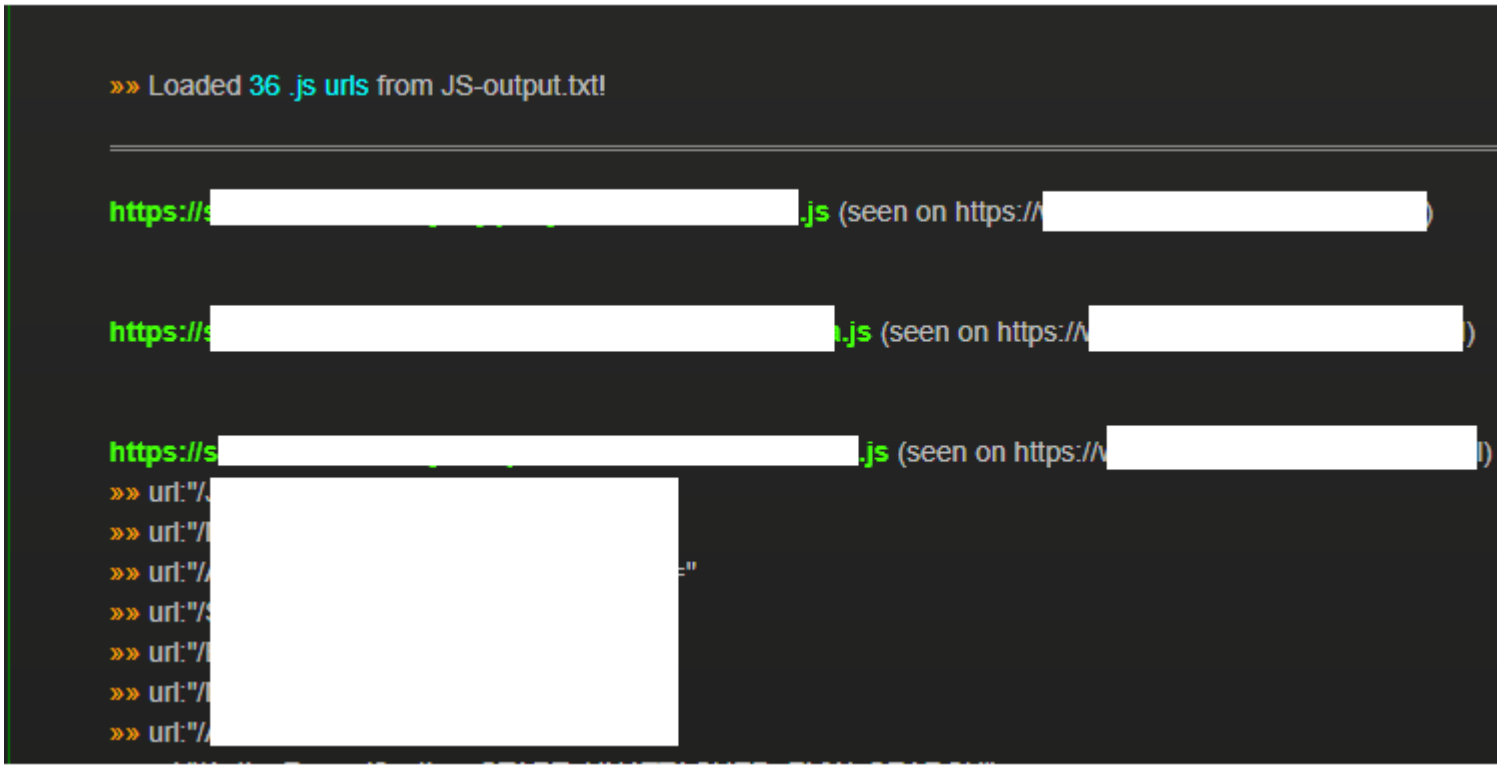
Find New Endpoints from JS Files (JS-Scan)

- Open JS-Scan in browser



Find New Endpoints from JS Files (JS-Scan)

- Click on Run Scanner and you will see something similar to this. That's it.



Takeaways

- Endpoints extracted from JS files are more vulnerable then Endpoints defined in WebPages.
- Automated Scanners generally don’t scan Endpoints defined in JS files.
- Developers & Testers don’t care about them.

Now We Have

WaybackURls

+

Subdomains

+

Subdomains of Subdomains

+

Other Sites on the same Domain

+

IP Range

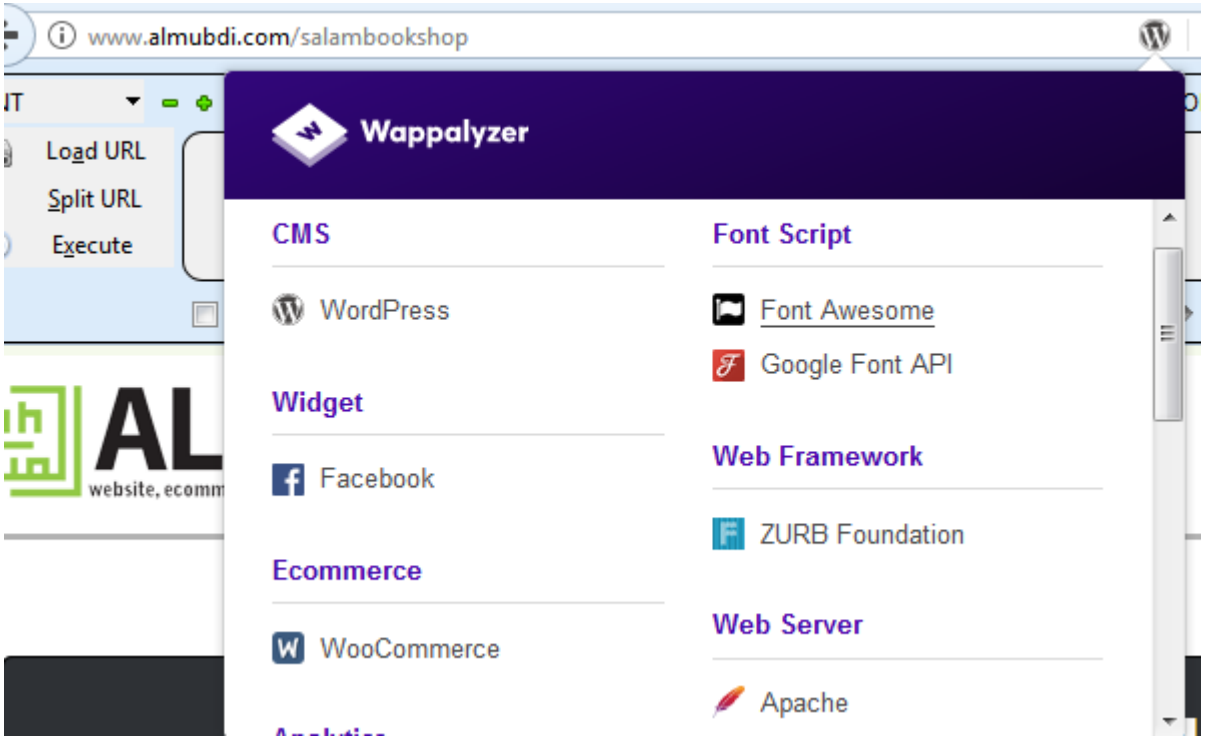
+

New Endpoints From JS FIles

Technologies Used by Web

Technologies Used by Web

- Wappalizer (Mozilla Addon)



Amazon Web Services

AWS or S3 Buckets

Amazon Web Services

- AWS Simple Storage Service (often shortened to S3) is used by companies that don't want to build and maintain their own storage repositories
- By using Amazon Simple Storage Service, they can store objects and files on a virtual server instead of on physical racks

Amazon Web Services

- After the user has created their bucket, they can start storing their source code, certificates, passwords, content, databases and other data.

Amazon Web Services

What if target is vulnerable

- You can get full access to S3 bucket
- You can download, upload and overwrite files.

How to find S3 Buckets?

Find S3 Buckets

- Google Dork

  site: amazonaws.com inurl: yahoo

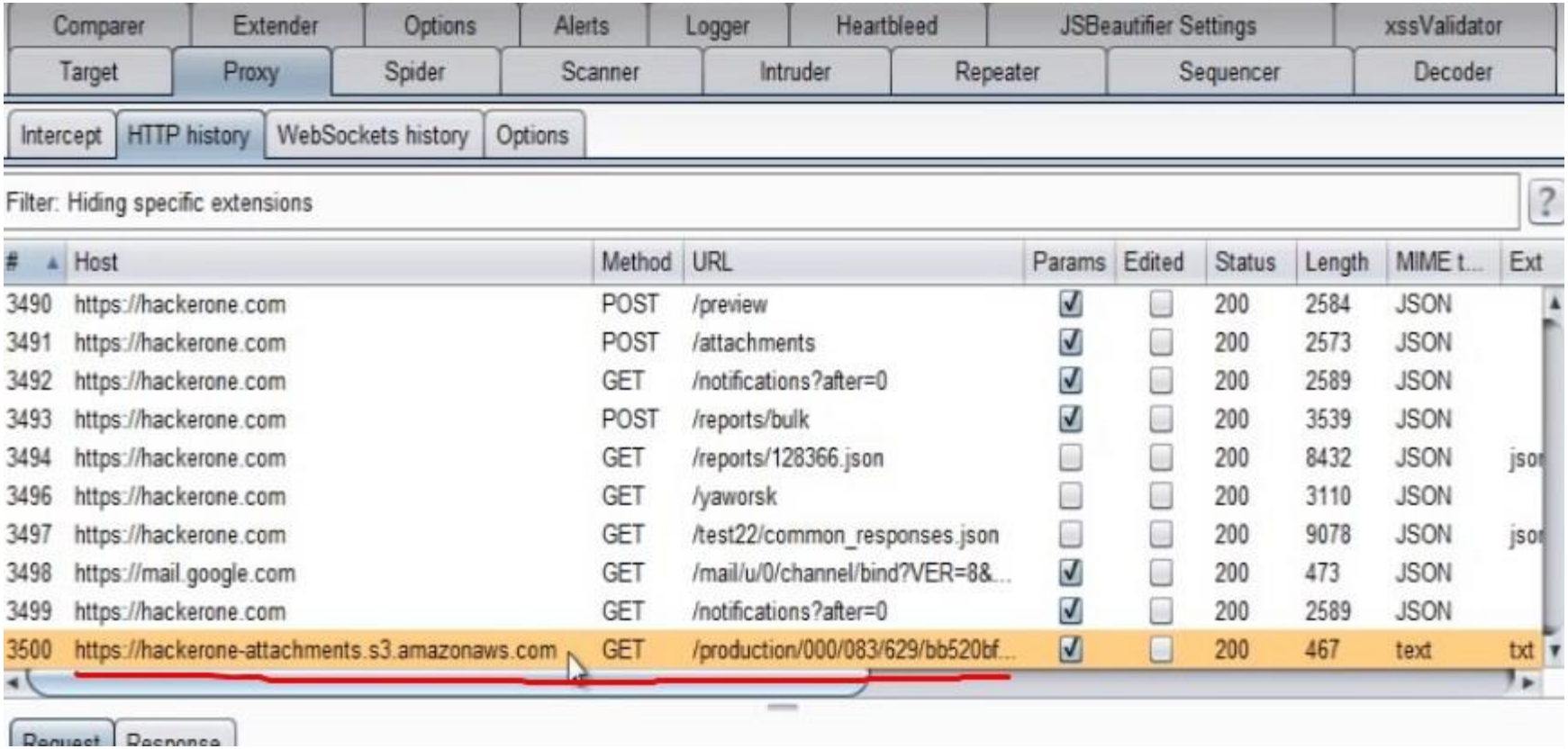
- Tool: S3 bucket finder

  (Download: https://digi.ninja/projects/bucket_finder.php)

```
./bucket_finder.rb my_words
```

Find S3 Buckets

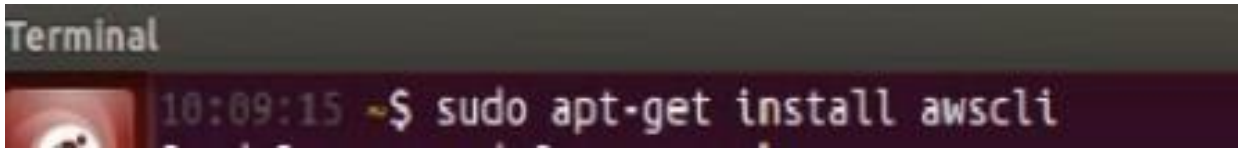
- Burp Suite can also Help



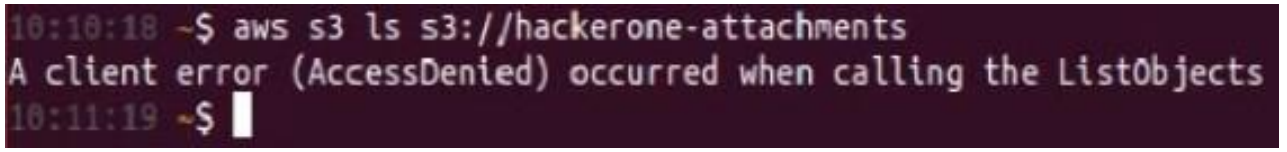
AWS
HACKING

AWS HACKING

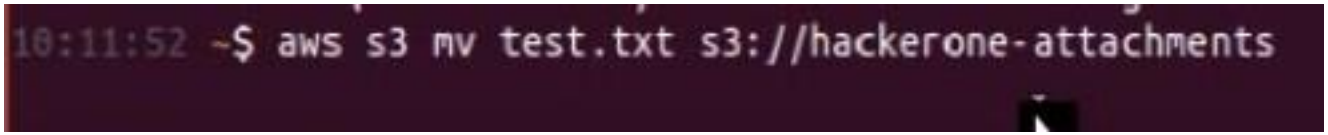
- Install awscli in kali



- Interact with Bucket



- Find World Writable Directory.



Now We Have

WaybackURls

+

Subdomains

+

Subdomains of Subdomains

+

Other Sites on the same Domain

+

IP Range

+

New Endpoints From JS Files

+

S3 Buckets

Github Recon

What you can find on Github

- FTP Credentials
- Secret Keys [API_key, Aws_secret key, etc.]
- Internal credentials [Employee credentials]
- API Endpoints
- Domain Patterns

Github Recon

- Go to github and search

Eg.

- “target.com” “dev”

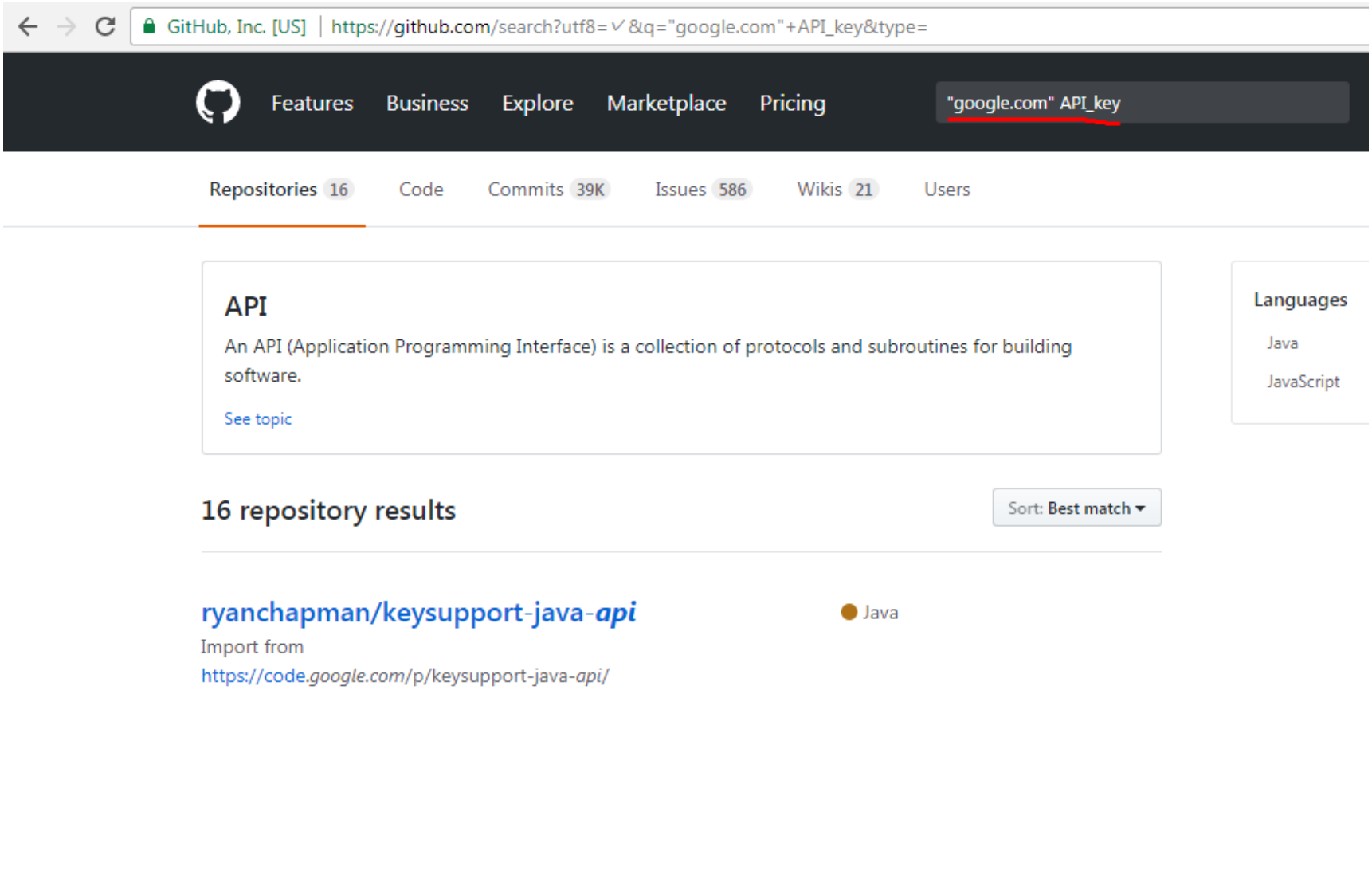
- “dev.target.com”

- “target.com” API_key

- “target.com” password

- “api.target.com”

Github Recon

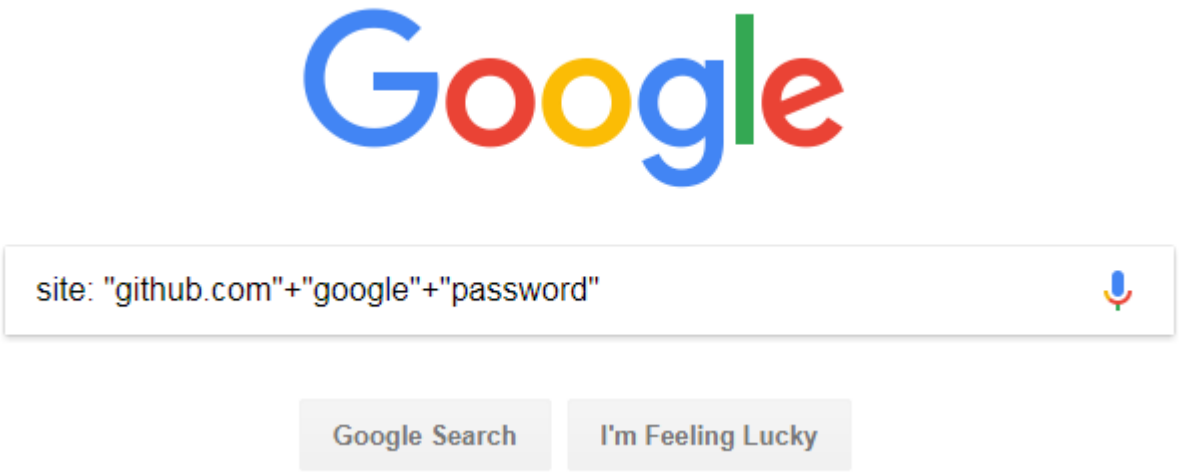


Github Recon

- Google can also help

Dork:

site: “github.com” + “Target” + password



Github Recon

Tools are out to automate this

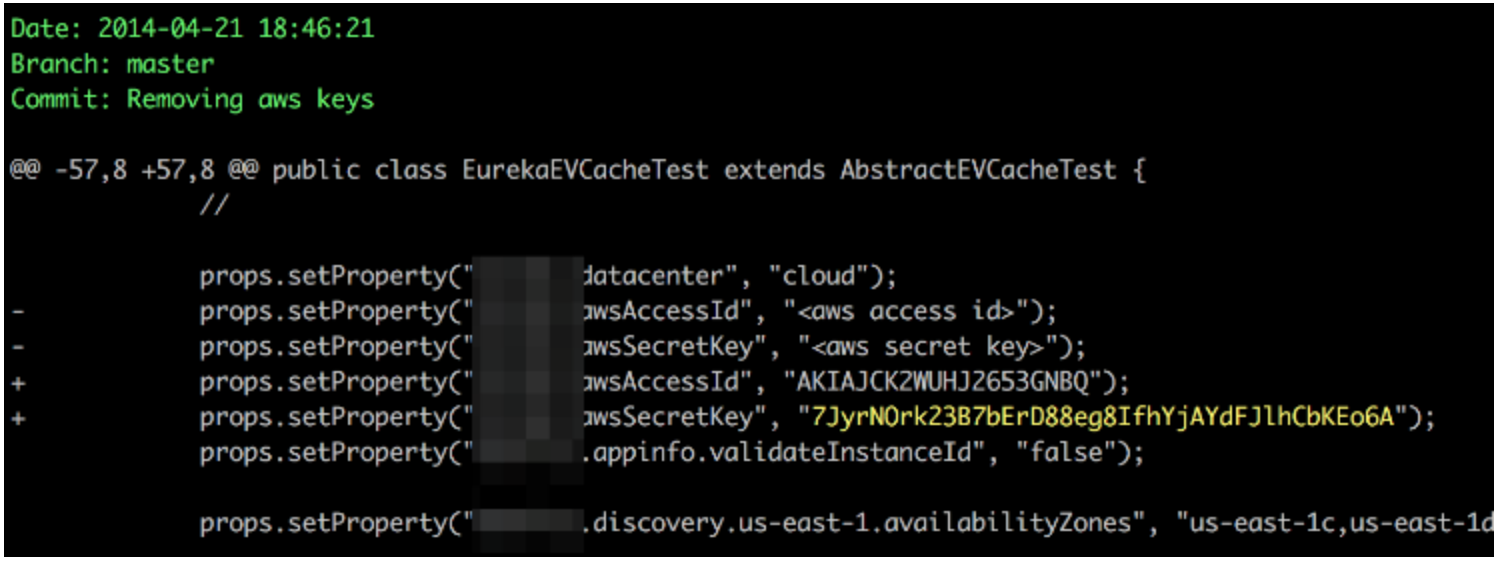
- Gitrob
- Git-all-secrets
- truffleHog
- Git-secrets
- Repo-supervisor
- Do it manually [Best way]

  – All tools are available on github

Tool- truffleHog

- Usage:

truffleHog --regex --entropy=False https://github.com/dxa4481/truffleHog.git



Content Discovery

Content Discovery

- Google is your friend
- Use Google Dork to find:-
  - File Extensions
  - Parameters
  - Login Page
  - Sometimes Directory Structure
  - Important Stuff

Content Discovery

- I often use Google Dork to find files with specific extension which also reveal technology used by Target.
- Google Dork:

  -site:target.com filetype:php

  - site:target.com filetype:aspx

  - site:target.com filetype:swf (Shockwave Flash)

  - site:target.com filetype:wsdl

Content Discovery

- Find Parameter
- Google Dork:

  - site: target.com inurl:.php?id=

  - site: target.com inurl:.php?user=

  - site: target.com inurl:.php?book=

Content Discovery

- Find Login Page
- Google Dork
  - site: target.com inurl:login.php
  - site: target.com intext: “login”
  - site: target.com inurl:portal.php
  - site: target.com inurl:register.php

(Note: if site has register page, there are chances that site also have login page)

Content Discovery

- Find Directory Structure
- Google Dork:

  -site: target.com intext: “index of /”

Content Discovery

- Find important Stuff
- Google Dork:

  -site: target.com filetype:txt

  - site: target.com inurl:.php.txt

  -site: target.com ext:txt

In most cases you will find robot.txt

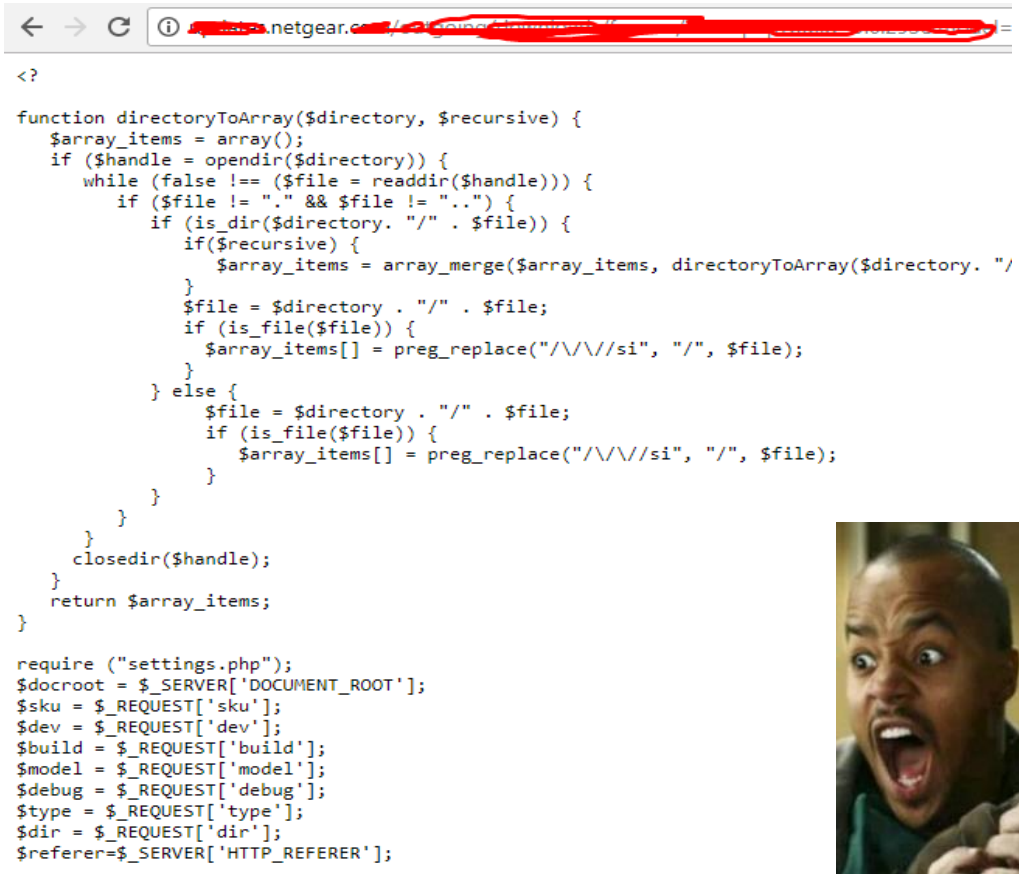
But sometimes you will find really juicy stuff

Content Discovery

- I found code in txt file which includes FTP credentials, SMTP credentials

Content Discovery

- Even some big names in IT Field.



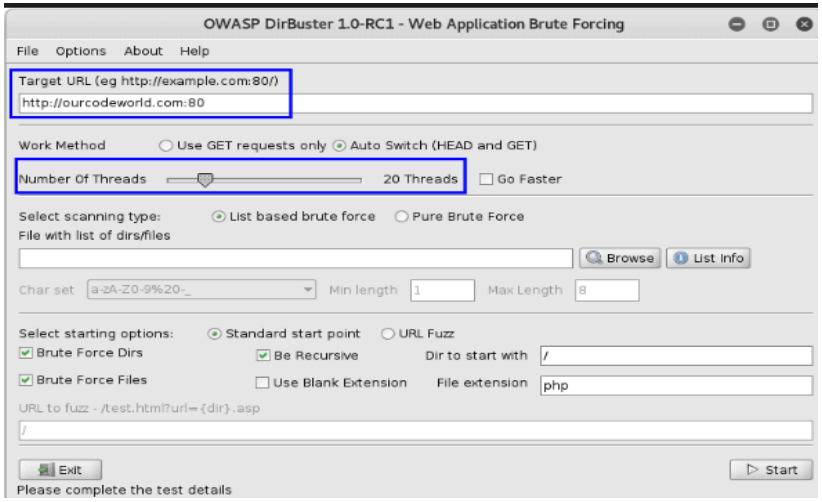
```
<?

function directoryToArray($directory, $recursive) {
   $array_items = array();
   if ($handle = opendir($directory)) {
      while (false !== ($file = readdir($handle))) {
         if ($file != "." && $file != "..") {
            if (is_dir($directory. "/" . $file)) {
               if($recursive) {
                  $array_items = array_merge($array_items, directoryToArray($directory. "/
               }
               $file = $directory . "/" . $file;
               if (is_file($file)) {
                 $array_items[] = preg_replace("/\/\//si", "/", $file);
               }
            } else {
                 $file = $directory . "/" . $file;
                 if (is_file($file)) {
                    $array_items[] = preg_replace("/\/\//si", "/", $file);
                 }
            }
         }
      }
     closedir($handle);
   }
   return $array_items;
}

require ("settings.php");
$docroot = $_SERVER['DOCUMENT_ROOT'];
$sku = $_REQUEST['sku'];
$dev = $_REQUEST['dev'];
$build = $_REQUEST['build'];
$model = $_REQUEST['model'];
$debug = $_REQUEST['debug'];
$type = $_REQUEST['type'];
$dir = $_REQUEST['dir'];
$referer=$_SERVER['HTTP_REFERER'];
```

Content Discovery

- Tools:
  - GoBuster [https://github.com/OJ/gobuster]

  Use:

  gobuster –w wordlist.txt –u http://trgt.com
  - Dirbuster



Thank You