

Recon Like A Boss



More Targets- More Options- More Opportunities



AGENDA

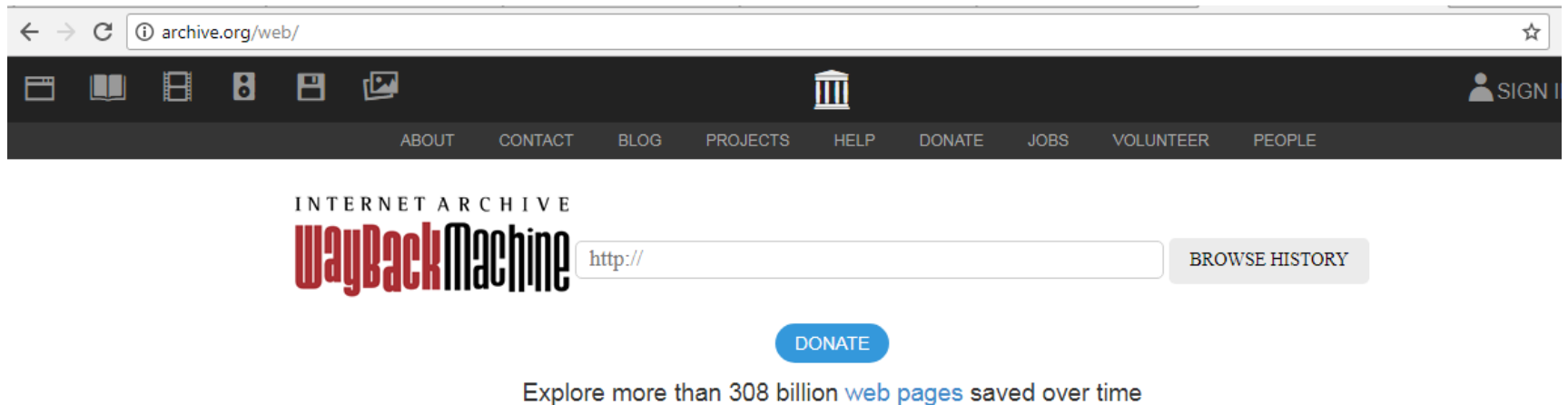
- Increase Your Attack Area
- Determine Technologies used by Website.
- Amazon Web Service (AWS) Recon & Hacking
- Github Recon
- Content Discovery

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`

```

[17] 4.1
[Knockpy]

+ checking for virustotal subdomains: YES
[
  "a.ns.hackerone.com",
  "b.ns.hackerone.com",
  "api.hackerone.com",
  "links.hackerone.com",
  "support.hackerone.com",
  "info.hackerone.com",
  "www.hackerone.com"
]
+ checking for wildcard: NO
+ checking for zonetransfer: NO
+ resolving target: YES
- scanning for subdomain...

Ip Address      Status  Type   Domain Name      Server
-----
162.159.0.31    301     host   a.ns.hackerone.com  cloudflare-nginx
104.16.99.52    301     host   api.hackerone.com   cloudflare-nginx
104.16.100.52   301     host   api.hackerone.com   cloudflare-nginx
162.159.1.31    301     host   b.ns.hackerone.com  cloudflare-nginx
104.16.12.26    301     host   support.hackerone.com cloudflare-nginx
104.16.13.26    301     host   support.hackerone.com cloudflare-nginx
104.16.99.52    301     host   www.hackerone.com   cloudflare-nginx
104.16.100.52   301     host   www.hackerone.com   cloudflare-nginx

```

Sub-domains Discovery

Sublist3r

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

```
[ahmed@secgeek ~/Sublist3r]$ python sublist3r.py -d yahoo.com
```

SUBLIST3R

Coded By Ahmed Aboul-Ela - @aboul3la

```
[+] Enumerating subdomains now for yahoo.com  
[-] Searching now in Baidu..  
[-] Searching now in Yahoo..  
[-] Searching now in Google..  
[-] Searching now in Bing..  
[-] Searching now in Ask..  
[-] Searching now in Netcraft..  
[-] Searching now in DNSdumpster..  
[-] Searching now in Virustotal..  
[-] Searching now in SSL Certificates..  
[-] Searching now in PassiveDNS..  
[-] Starting bruteforce module now using subbrute..  
[-] Total Unique Subdomains Found: 14015
```

Sub-domains Discovery

Sublist3r Cont.

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

```
File Edit View Bookmarks Settings Help
[ahmed@secgeek ~/Sublist3r]$ python sublist3r.py -d yahoo.com -b -t 50 -p 80,443

  SUBLIST3R

# Coded By Ahmed Aboul-Ela - @aboul3la

[-] Enumerating subdomains now for yahoo.com
[-] Searching now in Baidu..
[-] Searching now in Yahoo..
[-] Searching now in Google..
[-] Searching now in Bing..
[-] Searching now in Ask..
[-] Searching now in Netcraft..
[-] Searching now in DNSdumpster..
[-] Searching now in Virustotal..
[-] Searching now in SSL Certificates..
[-] Searching now in PassiveDNS..
[-] Starting bruteforce module now using subbrute..
[-] Total Unique Subdomains Found: 14015
[-] Start port scan now for the following ports: 80,443,21,22
1d.yahoo.com - Found open ports: 80
2010.yearinreview.yahoo.com - Found open ports: 80
```

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

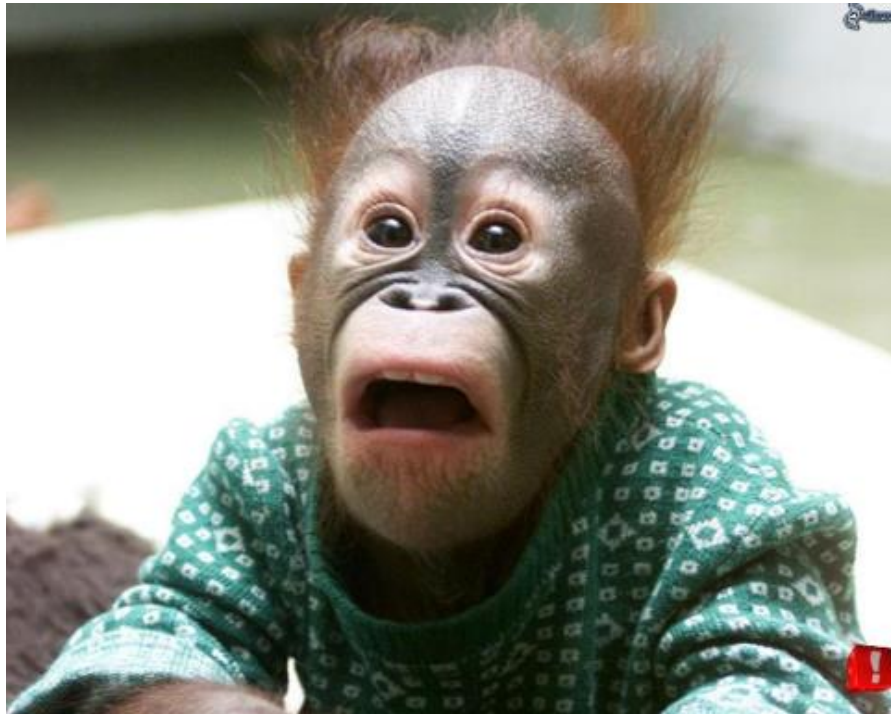
Don't Stop Here



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**

```
> ~/altdns ./altdns.py -i data/subdomains.txt -o april_output -w wordstest.txt -r -s resolved_results
[*] 500/48972 completed
[*] 1000/48972 completed
```

```
> ~/altdns cat resolved_results
acs.t... ..com:acs-... ..us-west-2.elb.amazonaws.com.
test. ....com:ec2-... ..us-west-1.compute.amazonaws.com.
apollo. ....com:.....
enigma. ....com:internal-... ..us-west-2.elb.amazonaws.com.
```

Find Subdomains of Subdomain

Tool: SubBrute

Usage:

```
./subbrute.py target.com > subdomains.txt
```

Then

```
./subbrute.py -t subdomains.txt
```

Now We Have

WaybackURLs

+

Subdomains

+

Subdomains of Subdomains

Sub-domain Validation

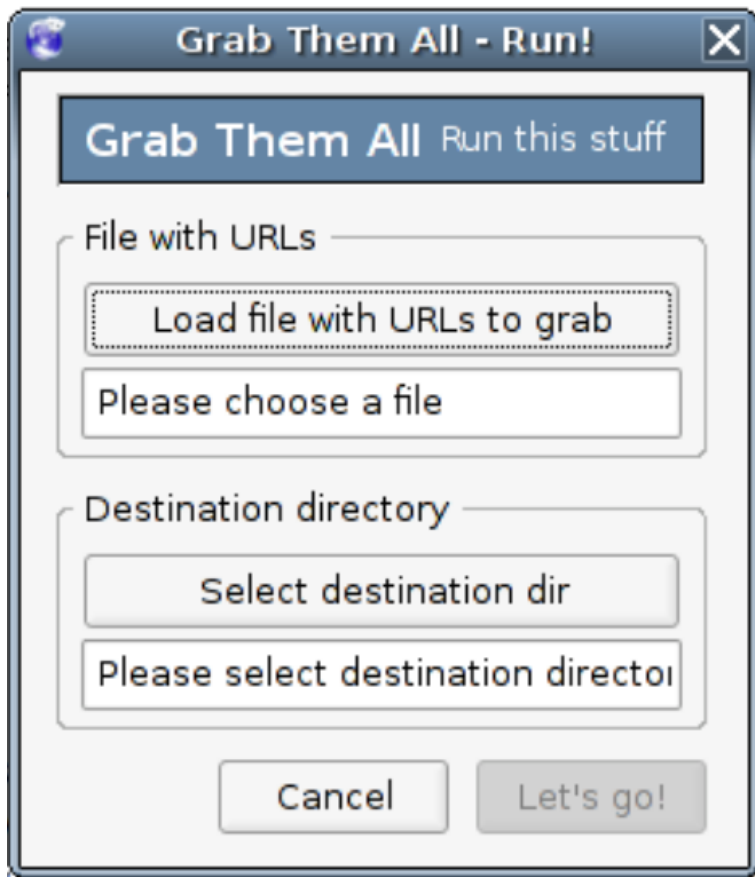
Tool: EyeWitness (<https://github.com/ChrisTruncer/EyeWitness>)

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

Reverse IP Domain Check

Remote Address

0.facebooko.com
0.facebook.co.id
0.facebook.it
ads.facebook.com
ar-ar.fb.me
autos.fb.com
baijee.tw
bingoblitz.fb.me
cafethu7.com
chat.fb.me
connect.facebook.com
covey.facebook.co
cyber.me.fb.me
dl.fb.me
en-gb.lt-lt.m.fb.me
eu-es.ar-ar.fb.me
facebook.com
facebook.fb.me
fb.com
fb.me
fbsbx.com
fma.fb.me
free.facebook.co
free.facebook.org

0.facebook.co
0.facebook.de
4g.fb.me
api.fb.me
as.fb.me
az-az.connect.facebook.com
basicdomain.co.uk
bn-in.fb.me
cdn.fb.me
claroideiastv.com.br.facebookproxy.com
connect.fb.me
cyber.fb.me
developers.cdn.fb.me
edge-star-mini-shv-01-lax3.facebook.com
en-ud.fb.me
evelopers.cdn.fb.me
facebook.com
facebook.zxc.pm
fb.me
fbcdn.net
feedback.facebook.com
free.facebook.com
free.facebook.co.za
free.fb.me

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

Secure | <https://whois.arin.net/rest/net/NET-98-136-0-0-1/pft?s=98.138.253.109>

ARIN
American Registry for Internet Numbers

SEARCH WhoisRWS 98.138.253.109
all requests subject to [terms of use](#)

NUMBER RESOURCES PARTICIPATE POLICIES FEES & INVOICES KNOWLEDGE ABOUT US

ARIN Online
enter

WHOIS-RWS

You searched for: 98.138.253.109

Network	
Net Range	<u>98.136.0.0 - 98.139.255.255</u>
CIDR	98.136.0.0/14
Name	A-YAHOO-US9
Handle	NET-98-136-0-0-1
Parent	NET98 (NET-98-0-0-0-0)
Net Type	Direct Allocation
Origin AS	
Organization	Yahoo! Inc. (YHOO) ✓

RELEVANT

- > [ARIN Who Terms of S](#)
- > [Report Wh](#)
- > [Whois-RW document](#)
- > [ARIN Tech Discussion](#)
- > [Sample st](#)

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.)

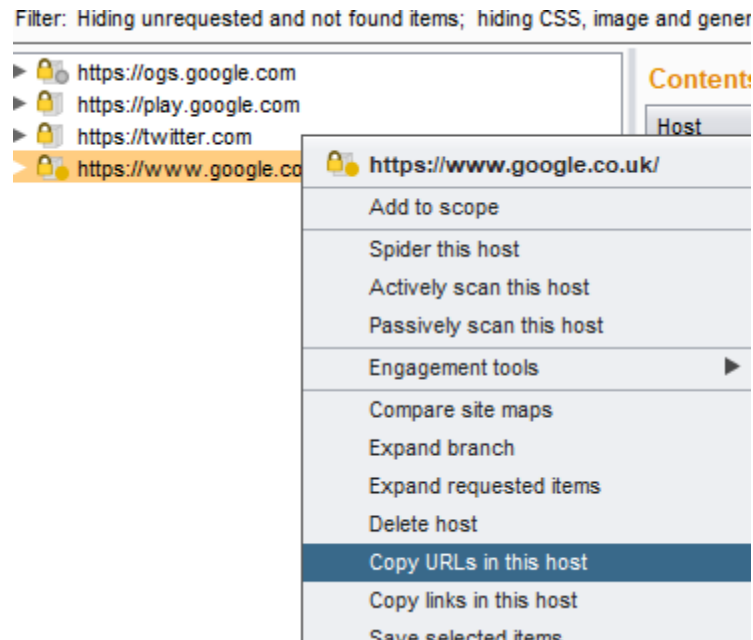
- Burp Suite: Proxy
- Zscanner: A tool designed to scrape a list of URLs. This tool will also scrape .js urls found on each page
- JS-Scan: A tool designed to scrape a list of .js files and extract urls

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

zScanner v1.0 by zseano

A tool designed to scrape a list of urls and extract all input names. Once extracted, payloads you define in payloads.txt will be appended to each parameter, then outputted for you to import into BURP. This scanner will also extract all .js files found.

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

A tool designed to scrape a list of .js urls and extract all urls found. You can modify the regex in the `processUrls()` function, which is located in this file. At the moment it just includes `url:"string"` and `url:'string'`.

Data is loaded from JS-output.txt in the root directory. You can use zScanner to scrape .js urls.

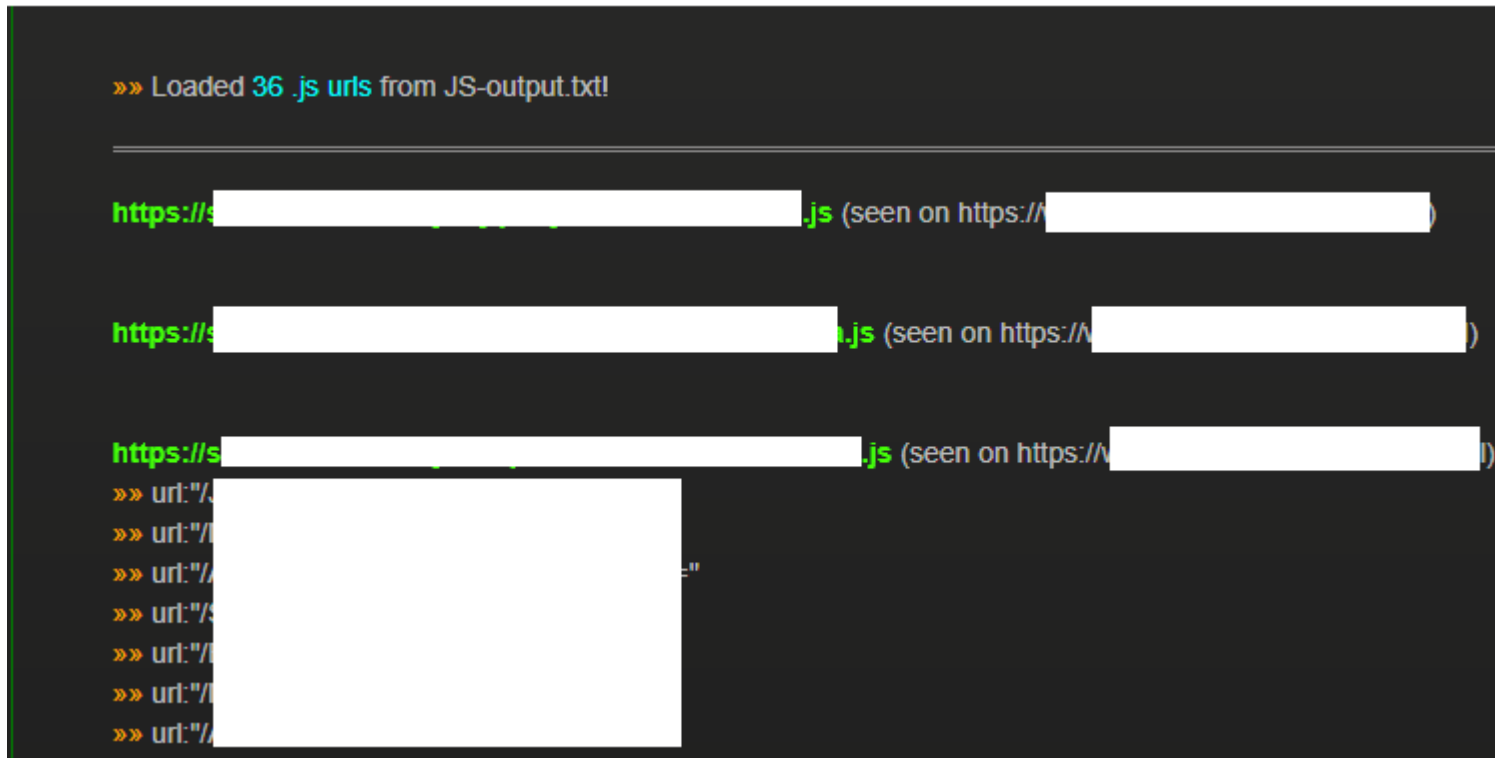
»» Loaded 36 .js urls from JS-output.txt!

Currently this script does not output anything, hence the visual view of urls found. You are free to modify this code to output how you want.

Run scanner

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 than 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

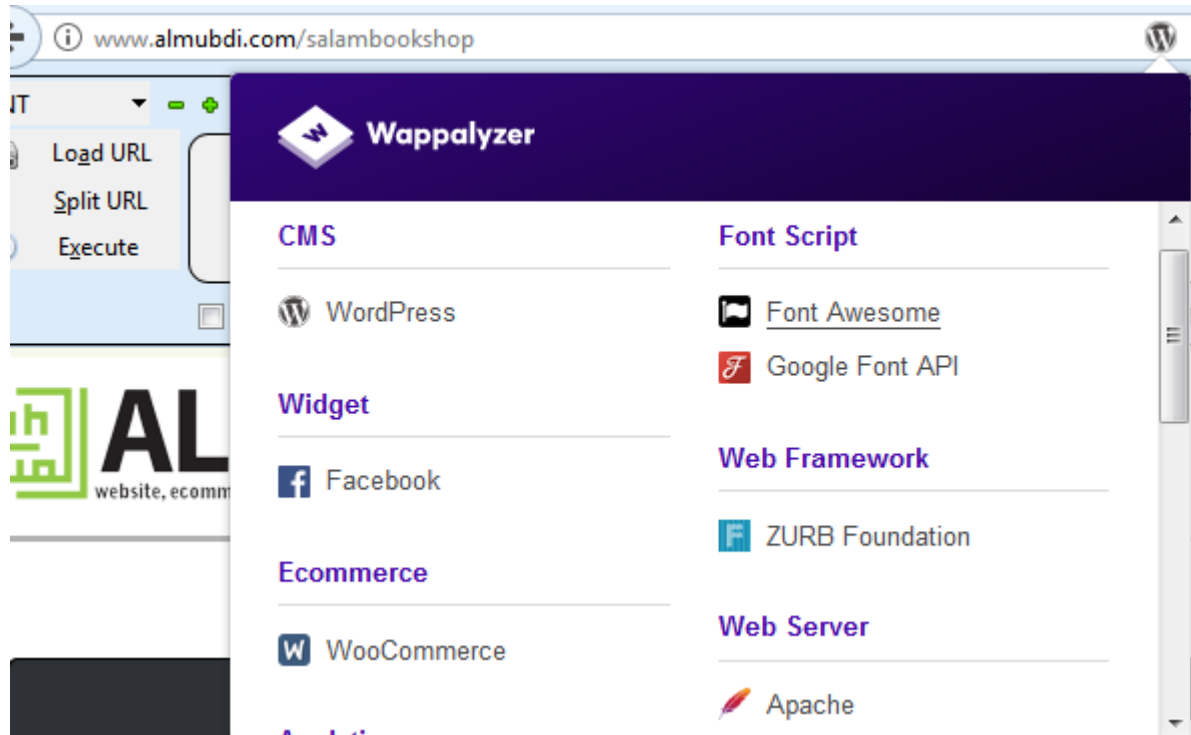
+

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

Filter: Hiding specific extensions

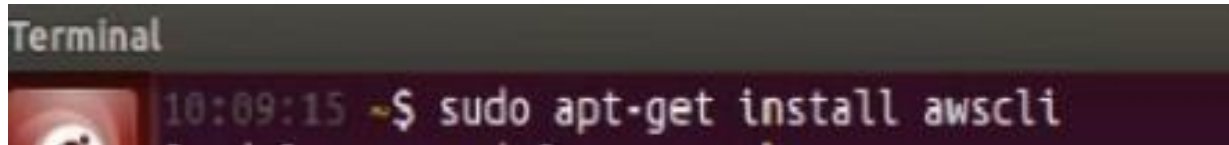
#	Host	Method	URL	Params	Edited	Status	Length	MIME t...	Ext
3490	https://hackerone.com	POST	/preview	<input checked="" type="checkbox"/>	<input type="checkbox"/>	200	2584	JSON	
3491	https://hackerone.com	POST	/attachments	<input checked="" type="checkbox"/>	<input type="checkbox"/>	200	2573	JSON	
3492	https://hackerone.com	GET	/notifications?after=0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	200	2589	JSON	
3493	https://hackerone.com	POST	/reports/bulk	<input checked="" type="checkbox"/>	<input type="checkbox"/>	200	3539	JSON	
3494	https://hackerone.com	GET	/reports/128366.json	<input type="checkbox"/>	<input type="checkbox"/>	200	8432	JSON	json
3496	https://hackerone.com	GET	/yaworsk	<input type="checkbox"/>	<input type="checkbox"/>	200	3110	JSON	
3497	https://hackerone.com	GET	/test22/common_responses.json	<input type="checkbox"/>	<input type="checkbox"/>	200	9078	JSON	json
3498	https://mail.google.com	GET	/mail/u/0/channel/bind?VER=8&...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	200	473	JSON	
3499	https://hackerone.com	GET	/notifications?after=0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	200	2589	JSON	
3500	https://hackerone-attachments.s3.amazonaws.com	GET	/production/000/083/629/bb520bf...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	200	467	text	txt

Request Response

AWS HACKING

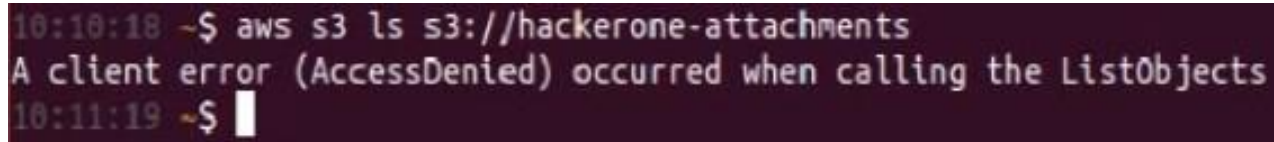
AWS HACKING

- Install awscli in kali

A terminal window titled "Terminal" with a dark background. The prompt is "10:09:15 ~\$" and the command "sudo apt-get install awscli" is entered.

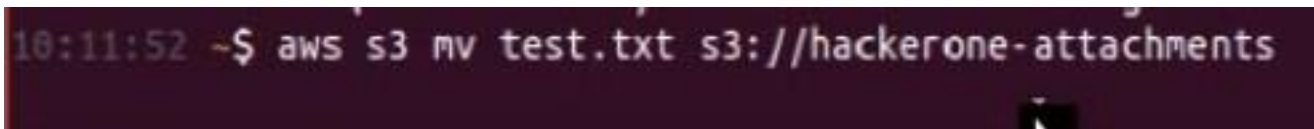
```
Terminal  
10:09:15 ~$ sudo apt-get install awscli
```

- Interact with Bucket

A terminal window with a dark background. The prompt is "10:10:18 ~\$" and the command "aws s3 ls s3://hackerone-attachments" is entered. The output is "A client error (AccessDenied) occurred when calling the ListObjects". The next prompt is "10:11:19 ~\$" with a cursor.

```
10:10:18 ~$ aws s3 ls s3://hackerone-attachments  
A client error (AccessDenied) occurred when calling the ListObjects  
10:11:19 ~$
```

- Find World Writable Directory.

A terminal window with a dark background. The prompt is "10:11:52 ~\$" and the command "aws s3 mv test.txt s3://hackerone-attachments" is entered.

```
10:11:52 ~$ aws s3 mv test.txt s3://hackerone-attachments
```

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

 [Features](#) [Business](#) [Explore](#) [Marketplace](#) [Pricing](#)

[Repositories 16](#) [Code](#) [Commits 39K](#) [Issues 586](#) [Wikis 21](#) [Users](#)

API

An API (Application Programming Interface) is a collection of protocols and subroutines for building software.

[See topic](#)

Languages

- Java
- JavaScript

16 repository results

Sort: Best match ▾

[ryanchapman/keysupport-java-api](#)

● Java

Import from

<https://code.google.com/p/keysupport-java-api/>

Github Recon

- Google can also help

Dork:

site: "github.com" + "Target" + password



site: "github.com"+"google"+"password"



Google Search

I'm Feeling Lucky

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>

```
Date: 2014-04-21 18:46:21
```

```
Branch: master
```

```
Commit: Removing aws keys
```

```
@@ -57,8 +57,8 @@ public class EurekaEVCacheTest extends AbstractEVCacheTest {  
    //
```

```
    props.setProperty("datacenter", "cloud");  
-    props.setProperty("awsAccessId", "<aws access id>");  
-    props.setProperty("awsSecretKey", "<aws secret key>");  
+    props.setProperty("awsAccessId", "AKIAJCK2WUHJ2653GNBQ");  
+    props.setProperty("awsSecretKey", "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");  
    props.setProperty("appinfo.validateInstanceId", "false");  
  
    props.setProperty("discovery.us-east-1.availabilityZones", "us-east-1c,us-east-1d
```

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 /"



A screenshot of a web browser window. The address bar shows the URL "sebastienguillon.com/test/php/". Below the address bar, the page title is "Index of /test/php". The main content area displays a directory listing table with four columns: Name, Last modified, Size, and Description. The table lists various PHP files and a parent directory link.

Name	Last modified	Size	Description
Parent Directory	-	-	-
date.php	29-Jan-2006 21:10	908	
file-upload.php	29-Jan-2006 21:10	1.0K	
file-upload.xhtml.php	29-Jan-2006 21:10	1.1K	
get_html_translation...>	29-Jan-2006 21:10	1.5K	
gettimeofday.php	29-Jan-2006 21:10	684	
host.php	06-Jun-2006 07:42	1.4K	
htmlentities.php	29-Jan-2006 21:10	1.1K	
htmlspecialchars.php	01-Apr-2006 16:40	1.8K	
optgroup.php	21-Jun-2006 04:33	1.8K	
prefixes-multiples-b...>	27-May-2006 03:57	13K	
test.php	10-Feb-2006 21:51	1.0K	
url-decode.php	20-Jan-2007 13:12	2.2K	

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



The image shows a web browser window with a secure connection (https://) and a code editor displaying C# code. The code includes various system namespaces and a class named Filtered, which contains private fields for API key, temp file, access token, and current position, along with public methods for initialization and authentication.

```
using System;
using System.Collections.Generic;
using System.Configuration;
using System.Globalization;
using System.IO;
using System.Linq;
using System.Net;
using System.Windows.Forms;
using CTCT;
using CTCT.Components;
using CTCT.Components.Contacts;
using EASendMail;
using PostmarkDotNet;
using Telerik.WinControls;
using Telerik.WinControls.UI;
using FilterCustomerList.com.securefreedom.api;

namespace FilterCustomerList
{
    public partial class Filtered : RadForm
    {
        private readonly string _apiKey = string.Empty;
        private readonly string tempFile = Path.GetTempFileName();
        private readonly string tempFileError = Path.GetTempFileName();
        private string _accessToken = string.Empty;
        private ConstantContact constantContact;
        private long currentPosition;
        public string errorPath;

        public Filtered()
        {
            InitializeComponent();
            _apiKey = ConfigurationManager.AppSettings["APIKey"];
        }

        public void authenticateUser()
        {
            adminLoginToolStripMenuItem.Visibility = ElementVisibility.Hidden;
            logOffAdminToolStripMenuItem.Visibility = ElementVisibility.Visible;
            chkConsultant.Enabled = true;
            chkCustomer.Enabled = true;
        }
    }
}
```

Content Discovery

- Even some big names in IT Field.

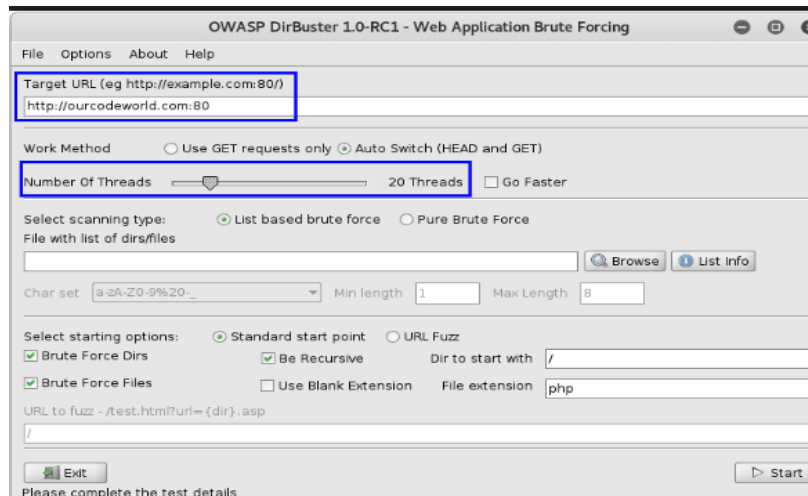
```
<?
function directoryToArray($directory, $recursive) {
    $array_items = array();
    if ($handle = opendir($directory)) {
        while (false !== ($file = readdir($handle))) {
            if ($file != "." && $file != "..") {
                if (is_dir($directory . "/" . $file)) {
                    if ($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