

hackerone

# It's the Little Things II

## Exploiting Vulnerabilities Through Proper Reconnaissance

Ben Sadeghipour (@Nahamsec)

# Intro



- Researcher, pentester, hacker, and bug bounty participant
- Hacker Operations Lead at HackerOne
- 600+ valid vulnerabilities to ~100 companies on HackerOne (Department of Defense, Airbnb, Oath/Yahoo, Snapchat, Valve, Zendesk, etc.)
- Bug Bounty Forum co-founder (@bugbountyforum)

# AGENDA

1. Overview
2. Asset Discovery
3. Content Discovery
4. Automation
5. Digital Dumpster Diving
6. Real Life Examples

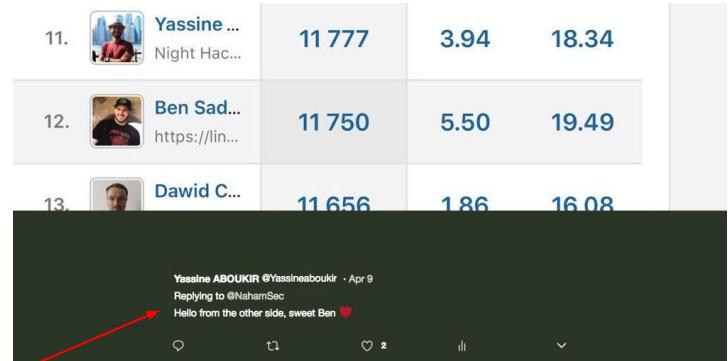
# Why?

- Self improvement
- Networking
- My career was built / boosted thanks to bug bounties



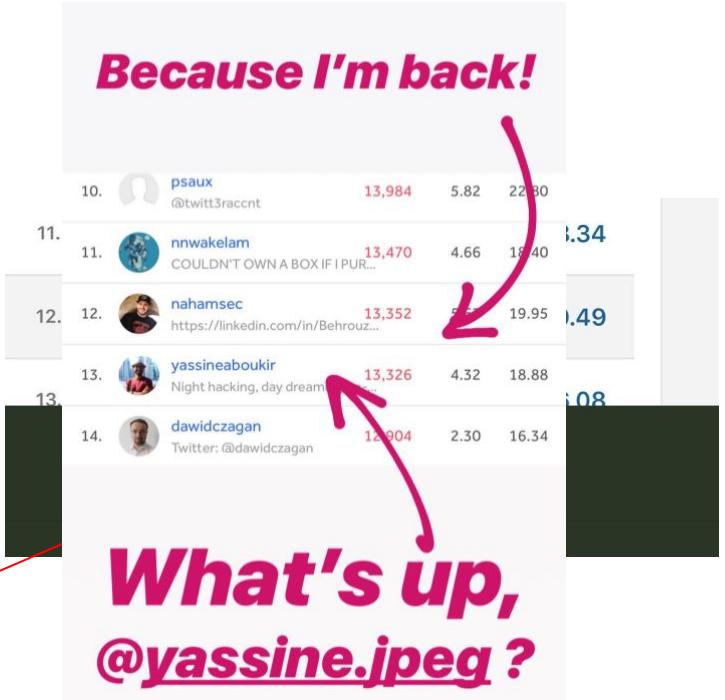
# Why?

- Self improvement
- Networking
- My career was built / boosted thanks to bug bounties
- Competition makes it more fun



# Why?

- Self improvement
- Networking
- My career was built / boosted thanks to bug bounties
- Competition makes it more fun



# Why?

- Self improvement
- Networking
- My career was built / boosted thanks to bug bounties
- Competition makes it more fun
- ... who doesn't like extra cash?



# Reconnaissance

# Recon (Definition)

In military operations, reconnaissance or scouting is the exploration outside an area occupied by friendly forces to gain information about natural features and other activities in the area.

# Recon (Definition)

- Understanding how the application is built
- Understanding how the application processes data
- Finding all possible “entry” points or company assets
- and finding as many files, folders, or endpoints



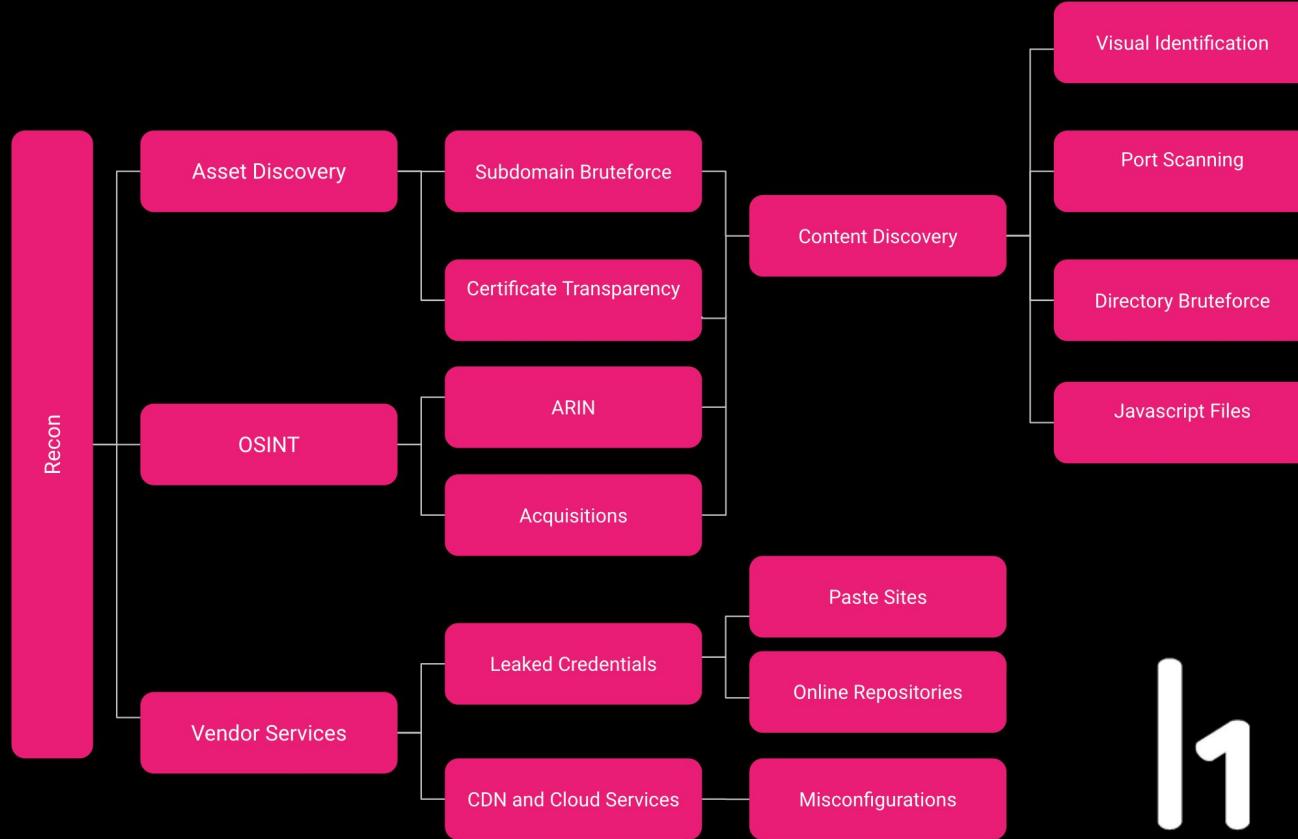
naffy

@nnwakelam

Following

I spend significantly more time doing recon than actually attacking anything.

# A Visual Guide to Recon

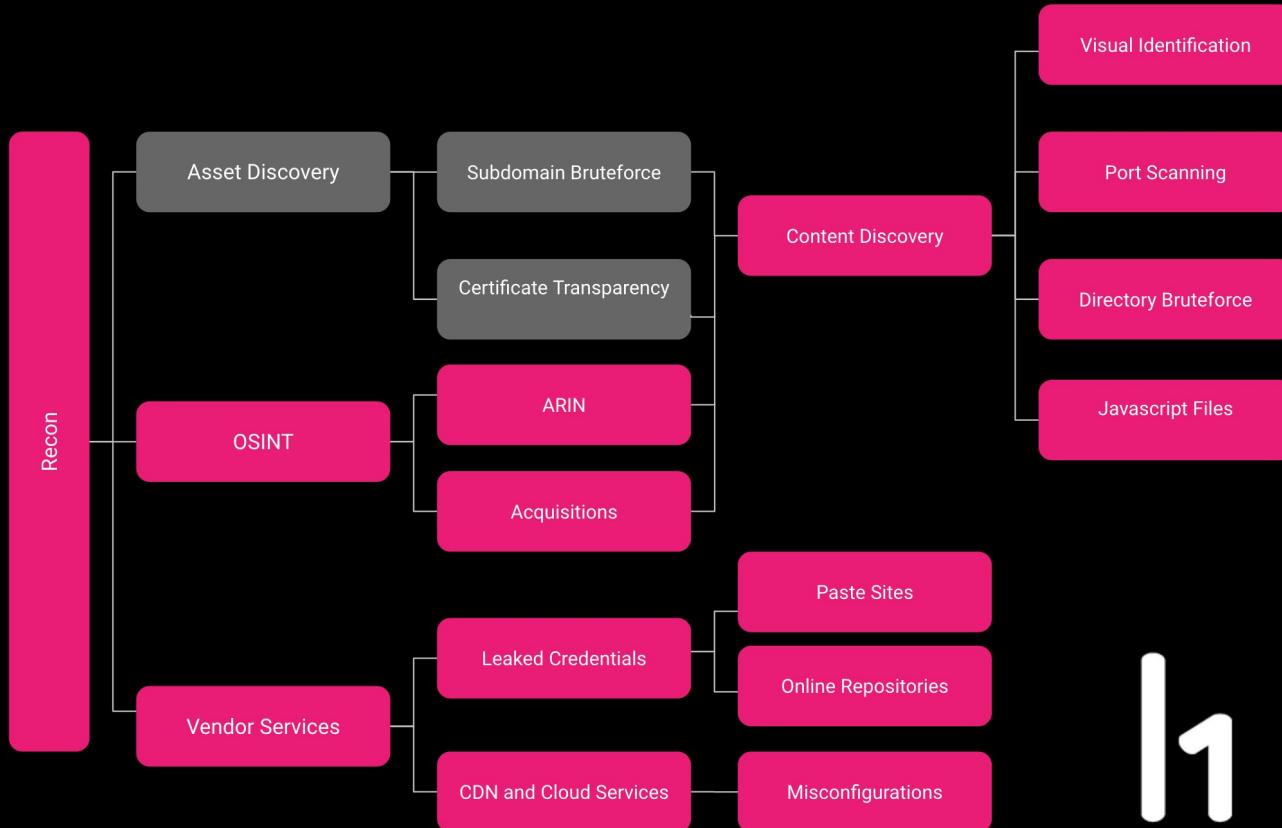


11

Ben  
Sadeghipour  
(@nahamsec)

# Asset Discovery

# A Visual Guide to Recon



1  
Ben  
Sadeghipour  
(@nahamsec)

# Asset Discovery

- Brute force
- Find different environments (.dev, .corp, .stage, uat, etc.)
- Brute force again
  - Different permutations
  - Different environment
    - dashboard.dev.site.com vs dashboard-dev.site.com

- sublist3r
- enumall
- massdns
- altdns
- brutesubs
- dns-parallel-prober
- dnscan
- knockpy
- tko-subs
- HostileSubBruteforce

Google Dork: site.com +inurl:dev -cdn

# Asset Discovery



- sublist3r
- enumall
- massdns
- altdns
- brutesubs
- dns-parallel-prober
- dnscan
- knockpy
- tko-subs
- HostileSubBruteforce

# Certificate Transparency

How do you find more?

# Censys

- Look for SSL certificates:
  - Example: 443.https.tls.certificate.parsed.extensions.subject\_alt\_name.dns\_names:snapchat.com

The screenshot shows the Censys search interface with the URL `https://censys.io/ipv4?q=443.https.tls.certificate.parsed.extensions.subject_alt_name.dns_names%3Asnapchat.com`. The search bar contains the query. The results page displays a list of IP addresses and their associated details. The results are as follows:

- 35.201.91.117 (117.91.201.35.bc.googleusercontent.com)
  - Google Inc. (15169) Ann Arbor, Michigan, United States
  - 443/https, 80/http, 8080/http
  - Error 404 (Page not found)!11 [snapshot-payments-gateway.snapchat.com](#)
  - Q 443.https.tls.chain.parsed.extensions.subject\_alt\_name.dns\_names: snapshot-payments-gateway.snapchat.com
- 35.186.197.135 (135.197.186.35.bc.googleusercontent.com)
  - Google Inc. (15169) Ann Arbor, Michigan, United States
  - 443/https, 80/http, 8080/http
  - Error 404 (Page not found)!11 [app-analytics.snapchat.com](#)
  - Q 443.https.tls.chain.parsed.extensions.subject\_alt\_name.dns\_names: app-analytics.snapchat.com
- 35.186.226.184 (184.226.186.35.bc.googleusercontent.com)
  - Google Inc. (15169) Ann Arbor, Michigan, United States
  - 443/https, 80/http, 8080/http
  - Error 404 (Page not found)!11 [tr.snapchat.com](#)
  - Q 443.https.tls.chain.parsed.extensions.subject\_alt\_name.dns\_names: tr.snapchat.com
- 35.201.121.17 (17.121.201.35.bc.googleusercontent.com)
  - Google Inc. (15169) Ann Arbor, Michigan, United States
  - 443/https, 80/http, 8080/http
  - Error 404 (Page not found)!11 [app-analyticsv2.snapchat.com](#)
  - Q 443.https.tls.certificate.parsed.extensions.subject\_alt\_name.dns\_names: app-analyticsv2.snapchat.com
- 35.186.236.63 (63.236.186.35.bc.googleusercontent.com)
  - Google Inc. (15169) Ann Arbor, Michigan, United States
  - 443/https, 80/http, 8080/http
  - Error 404 (Page not found)!11 [staging.app-analytics.snapchat.com](#)
  - Q 443.https.tls.certificate.parsed.extensions.subject\_alt\_name.dns\_names: staging.app-analytics.snapchat.com

# Shodan

- Search by hostname
- Filter for
  - Ports: 8443, 8080, 8180, etc
  - Title: “Dashboard [Jenkins]”
  - Product:Tomcat
  - Hostname: somecorp.com
  - Org: evilcorp
  - ssl: Google



**Sam Curry (z1z)** @samwcyo · Aug 14

I'm only going to college for the free Shodan membership.

4

2

50



The screenshot shows the Shodan search interface with the query "org:airbnb port:8443". The results page displays a world map titled "TOP COUNTRIES" with data points for Singapore, United States, and other locations. A "Bad Request" section shows a log entry for a failed SSL connection. The "SSL Certificate" section details a certificate issued by DigiCert SHA2 Secure to Airbnb, Inc. over a DigiCert Inc server. The certificate is valid until 2018-09-03. The "Supported SSL Versions" section lists TLSv1.2. The "Diffie-Hellman Parameters" section shows a fingerprint for RFC5114/2048-bit MODP Group with 256-bit Prime Order Subgroup.

Shodan Developers Book View All...

SHODAN org:airbnb port:8443

Explore Downloads Reports Developer Pricing Enterprise Access Contact Us Show API Key Help Center My Account

Exploits Maps Share Search Download Results Create Report

**Bad Request**

8.18.221.51  
esv1-uc-mresd1.airgo.net  
Airbnb  
Added on 2018-09-03 19:44:22 GMT  
United States  
Details

**SSL Certificate**

HTTP/1.1 400 Bad Request  
Date: Fri, 03 Aug 2018 19:44:22 GMT  
Connection: keep-alive  
Server: CE\_Engine  
Cache-Control: no-store  
Content-Type: text/html  
Content-Language: en  
Content-Length: 65

**Supported SSL Versions**

TLSv1.2

**Diffie-Hellman Parameters**

Fingerprint: RFC5114/2048-bit  
MODP Group with 256-bit Prime Order  
Subgroup

# Certsspotter

- Great API
  - Easy to automate:
    - Make a bash alias
    - Automate
    - Win

root@vps82152:~# certspotter yahoo.com  
170prd.fin.yahoo.com  
2013-en-imagenes.es.yahoo.com  
360.mail.yahoo.com  
3arrebni.yahoo.com  
7-eleven.yahoo.com  
aa.lrd.yahoo.com  
about.yahoo.com  
abumediadomobile.query.yahoo.com  
abumedia.yahoo.com  
abumedia.yql.yahoo.com  
abuse.corp.yahoo.com  
abuse.yahoo.com  
academy-delivery.cc.corp.yahoo.com  
accmgr.secure.webhosting.yahoo.com  
accmgr.webhosting.yahoo.com  
accountkey.yahoo.com  
accountlink.www.yahoo.com  
accountlink.yahoo.com  
accountservic.corp.yahoo.com  
ace.ysm.yahoo.com  
aclpushdb.ops.yahoo.com  
a-cms.shp.corp.tw1.yahoo.com  
actapi.corp.yahoo.com  
actualites.yahoo.com  
adbuilder.creative.yahoo.com  
add.my.yahoo.com  
address.yahoo.com  
adlatencyvendoroutreach.yahoo.com  
admanagerplus.yahoo.com  
admanager.yahoo.com  
admetricsclone.udapp.yahoo.com  
admetrics.udapp.yahoo.com  
adminapp.creatr.corp.yahoo.com  
admin.bb.abuse.yahoo.com  
admin.bf1.yhs.search.yahoo.com  
admin.ckms.yahoo.com  
adminincms.corp.yahoo.com  
adminincms.labs.yahoo.com

# Certsspotter

- Great API
  - Easy to automate:
    - Make a bash alias
    - Automate
    - Win

# We'll get to this later

root@vps82152:~# certspotter yahoo.com  
170prd.fin.yahoo.com  
2013-en-imagenes.es.yahoo.com  
360.mail.yahoo.com  
3arrebni.yahoo.com  
7-eleven.yahoo.com  
aa.lrd.yahoo.com  
about.yahoo.com  
abumediadomobile.query.yahoo.com  
abumedia.yahoo.com  
abumedia.yql.yahoo.com  
abuse.corp.yahoo.com  
abuse.yahoo.com  
academy-delivery.cc.corp.yahoo.com  
accmgr.secure.webhosting.yahoo.com  
accmgr.webhosting.yahoo.com  
accountkey.yahoo.com  
accountlink.www.yahoo.com  
accountlink.yahoo.com  
accountservic.corp.yahoo.com  
ace.ysm.yahoo.com  
aclpushdb.ops.yahoo.com  
a-cms.shp.corp.tw1.yahoo.com  
actapi.corp.yahoo.com  
actualites.yahoo.com  
adbuilder.creative.yahoo.com  
add.my.yahoo.com  
address.yahoo.com  
adlatencyvendoroutreach.yahoo.com  
admanagerplus.yahoo.com  
admanager.yahoo.com  
admetricsclone.udapp.yahoo.com  
admetrics.udapp.yahoo.com  
adminapp.creatr.corp.yahoo.com  
admin.bb.abuse.yahoo.com  
admin.bf1.yhs.search.yahoo.com  
admin.ckms.yahoo.com  
adminincms.corp.yahoo.com  
adminincms.labs.yahoo.com

# Crt.sh

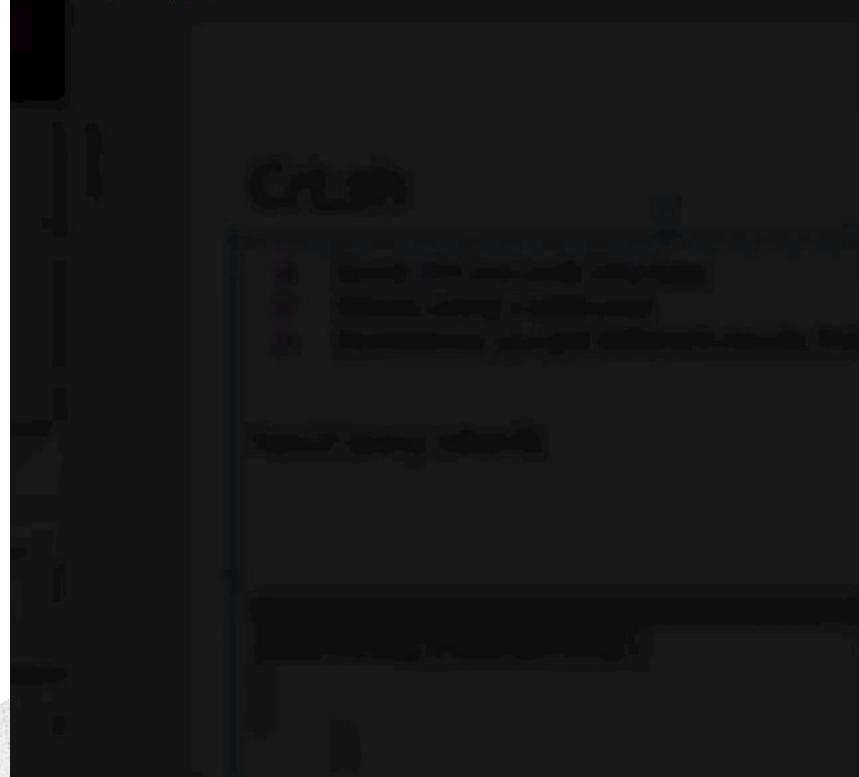
- Great API and web interface
- Allows using a wild card
- Sometimes you get different results from differents sources

Search query: **Identity LIKE '[www.snapchat.%](#)'**

crt.sh ID	Logged At	Not Before	Not After	Identity
<a href="#">771398392</a>	2018-09-20	2018-09-20	2018-12-19	<a href="#">www.snapchat.pizza</a>
<a href="#">771124506</a>	2018-09-20	2018-09-20	2018-12-19	<a href="#">www.snapchat.sale</a>
<a href="#">771122320</a>	2018-09-20	2018-09-20	2018-12-19	<a href="#">www.snapchat.properties</a>
<a href="#">771099401</a>	2018-09-20	2018-09-20	2018-12-19	<a href="#">www.snapchat.productions</a>
<a href="#">765647951</a>	2018-09-19	2018-09-18	2018-12-17	<a href="#">www.snapchat.timurmuhendislik.com</a>
<a href="#">763811249</a>	2018-09-18	2018-09-18	2018-12-17	<a href="#">www.snapchat.achimhepp.com</a>
<a href="#">759812331</a>	2018-09-17	2018-09-17	2018-12-16	<a href="#">www.snapchat.fashion.blog</a>
<a href="#">751920069</a>	2018-09-15	2018-09-15	2018-12-14	<a href="#">www.snapchat.easysteenvids.com</a>
<a href="#">751920069</a>	2018-09-15	2018-09-15	2018-12-14	<a href="#">www.snapchat.pink</a>

<https://crt.sh/?q=www.snapchat.%>

```
root@reconpad:~# crtsh facebook.com | wc -l  
136  
root@reconpad:~# [ ]
```

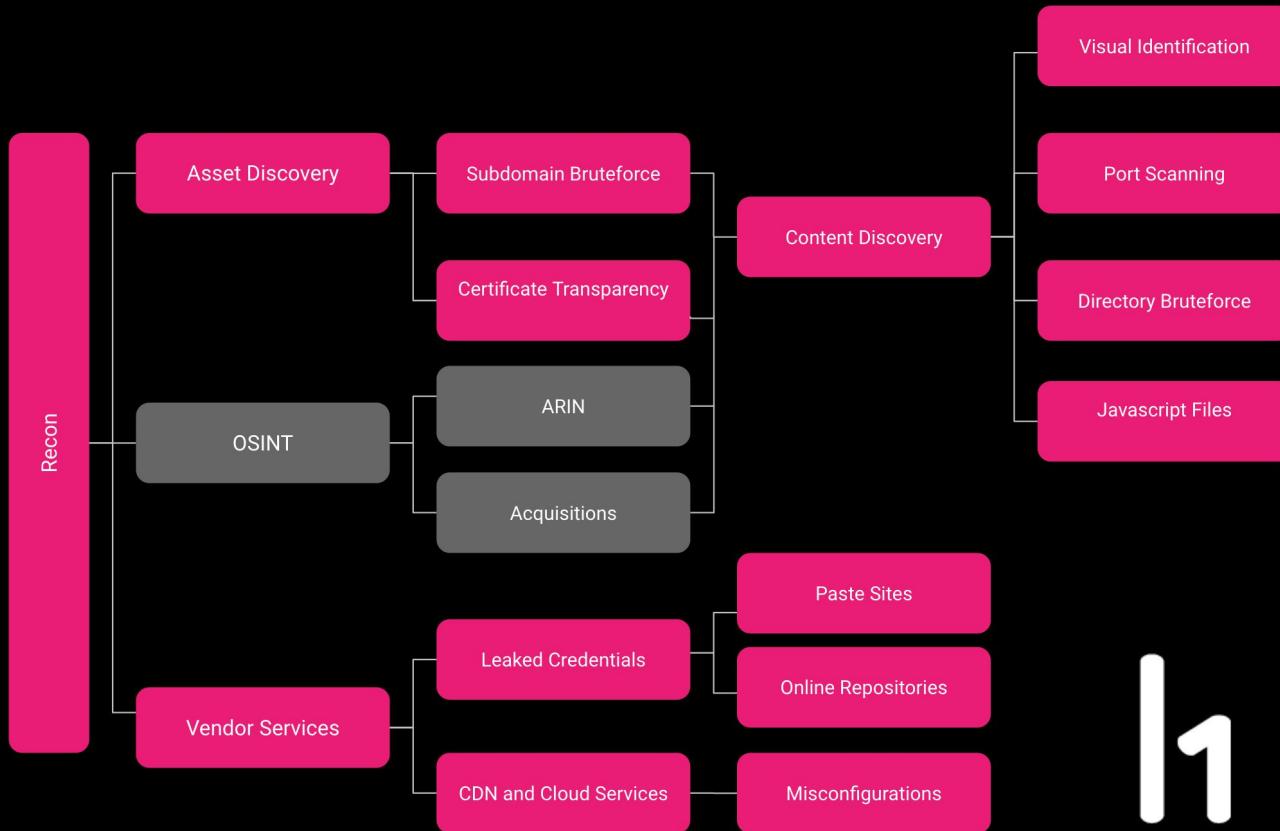


```
$ curl 'https://crt.sh/?q=%,$1'&output=json' | jq '.name_value' | sed 's/\\"//g' | sed 's/\^\\//g'
```

# OSINT

What about other properties?

# A Visual Guide to Recon



h1

Ben  
Sadeghipour  
(@nahamsec)

# Acquisitions

- Usually in scope (after 6 months)
- Geared towards bigger companies: Facebook, Google, Yahoo, etc.

The screenshot shows the Crunchbase website interface. At the top, there's a navigation bar with links for 'Products', 'Marketplace', 'News', and 'About'. A search bar is located at the top right. Below the navigation, there's a banner for 'Introducing Crunchbase Pro' with a 'LEARN MORE' button. On the left, there's a sidebar with various search filters and links: 'Companies', 'People', 'Investors', 'Funding Rounds', 'Acquisitions' (which is currently selected), 'Schools', 'Events', 'Hubs', 'My Searches', 'My Lists', 'Marketplace', and 'Add New Profile'. The main content area is titled 'Yahoo > Acquisitions' and contains a table with 21 rows, each representing an acquisition. The columns in the table are 'Acquired Organization Name', 'Announced Date', 'Price', and 'Transaction Name'. Each row includes a small thumbnail icon of the acquired company and a link to its profile. The acquisitions listed are: Polyvore (Jul 31, 2015, \$230M, Polyvore acquired by Ya...), MEDIA GROUP ONE (Jul 24, 2015, \$23M, MEDIA GROUP ONE acq...), Cooliris (Nov 21, 2014, -, Cooliris acquired by Yahoo), BrightRoll (Nov 11, 2014, \$640M, BrightRoll acquired by Ya...), MessageMe (Oct 3, 2014, \$30M, MessageMe acquired by ...), Bookpad (Sep 22, 2014, -, Bookpad acquired by Ya...), Luminate (Sep 7, 2014, -, Luminate acquired by Ya...), ClarityRay (Aug 15, 2014, -, ClarityRay acquired by Y...), Zofari (Aug 12, 2014, -, Zofari acquired by Yahoo), Flurry (Jul 21, 2014, \$200M, Flurry acquired by Yahoo), RayV (acquired by Yahoo!) (Jul 11, 2014, -, RayV (acquired by Yahoo...)), Blink Messenger (May 13, 2014, -, Blink Messenger acquire...), Vizify (Mar 5, 2014, -, Vizify acquired by Yahoo), Distill (Feb 13, 2014, -, Distill acquired by Yahoo), Wander (Feb 11, 2014, -, Wander acquired by Yahoo), Incredible Labs (Jan 30, 2014, -, Incredible Labs acquired ...), Tomfoolery (Jan 28, 2014, -, Tomfoolery acquired by ...), and Cloud Party (Jan 24, 2014, -, Cloud Party acquired by ...).

Acquired Organization Name	Announced Date	Price	Transaction Name
Polyvore	Jul 31, 2015	\$230M	Polyvore acquired by Ya...
MEDIA GROUP ONE	Jul 24, 2015	\$23M	MEDIA GROUP ONE acq...
Cooliris	Nov 21, 2014	-	Cooliris acquired by Yahoo
BrightRoll	Nov 11, 2014	\$640M	BrightRoll acquired by Ya...
MessageMe	Oct 3, 2014	\$30M	MessageMe acquired by ...
Bookpad	Sep 22, 2014	-	Bookpad acquired by Ya...
Luminate	Sep 7, 2014	-	Luminate acquired by Ya...
ClarityRay	Aug 15, 2014	-	ClarityRay acquired by Y...
Zofari	Aug 12, 2014	-	Zofari acquired by Yahoo
Flurry	Jul 21, 2014	\$200M	Flurry acquired by Yahoo
RayV (acquired by Yahoo!)	Jul 11, 2014	-	RayV (acquired by Yahoo...)
Blink Messenger	May 13, 2014	-	Blink Messenger acquire...
Vizify	Mar 5, 2014	-	Vizify acquired by Yahoo
Distill	Feb 13, 2014	-	Distill acquired by Yahoo
Wander	Feb 11, 2014	-	Wander acquired by Yahoo
Incredible Labs	Jan 30, 2014	-	Incredible Labs acquired ...
Tomfoolery	Jan 28, 2014	-	Tomfoolery acquired by ...
Cloud Party	Jan 24, 2014	-	Cloud Party acquired by ...

# ARIN

You searched for: Yahoo

**Customers**

- Yahoo (C00146168)
- Yahoo (C00146169)
- Yahoo (C01196389)

**Organizations**

- Yahoo (YAHOO-1)
- YAHOO (YAHOO-10)

# ARIN

Organization	
Name	Yahoo
Handle	YAHOO-1
Street	701 First Avenue
City	Sunnyvale
State/Province	CA
Postal Code	94089
Country	US
Registration Date	1996-10-14
Last Updated	2013-04-02
Comments	
RESTful Link	<a href="https://whois.arin.net/rest/org/YAHOO-1">https://whois.arin.net/rest/org/YAHOO-1</a>
See Also	<a href="#">Related networks.</a>
See Also	<a href="#">Related autonomous system numbers.</a>
See Also	<a href="#">Related POC records.</a>

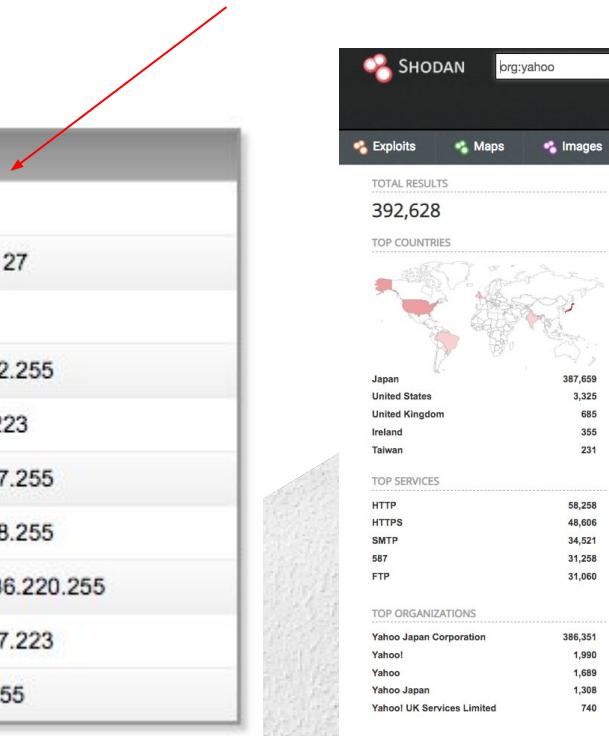
# ARIN

Network Resources	
LVLT-YAHOO-1-8-3-34 (NET-8-3-34-0-1)	8.3.34.0 - 8.3.35.255
NET-216-34-77-0 (NET-216-34-77-0-1)	216.34.77.0 - 216.34.77.127
COLOC-1-YAHOO-1-8-8-178 (NET-8-8-178-0-1)	8.8.178.0 - 8.8.178.255
SAVV-64-209-232-0-0-1 (NET-64-209-232-0-1)	64.209.232.0 - 64.209.232.255
SAVV-S235114-9 (NET-64-39-38-208-1)	64.39.38.208 - 64.39.38.223
SAVV-S235114-11 (NET-204-71-177-0-1)	204.71.177.0 - 204.71.177.255
SAVV-S235114-12 (NET-204-71-188-0-1)	204.71.188.0 - 204.71.188.255
SAVV-S235114-17 (NET-216-136-220-128-1)	216.136.220.128 - 216.136.220.255
SAVV-S235114-21 (NET-64-56-197-208-1)	64.56.197.208 - 64.56.197.223
LVLT-YAHOO-1-67-72-118 (NET-67-72-118-0-1)	67.72.118.0 - 67.72.119.255

# ARIN

Network Resources	
LVLT-YAHOO-1-8-3-34 (NET-8-3-34-0-1)	8.3.34.0 - 8.3.35.255
NET-216-34-77-0 (NET-216-34-77-0-1)	216.34.77.0 - 216.34.77.127
COLOC-1-YAHOO-1-8-8-178 (NET-8-8-178-0-1)	8.8.178.0 - 8.8.178.255
SAVV-64-209-232-0-0-1 (NET-64-209-232-0-1)	64.209.232.0 - 64.209.232.255
SAVV-S235114-9 (NET-64-39-38-208-1)	64.39.38.208 - 64.39.38.223
SAVV-S235114-11 (NET-204-71-177-0-1)	204.71.177.0 - 204.71.177.255
SAVV-S235114-12 (NET-204-71-188-0-1)	204.71.188.0 - 204.71.188.255
SAVV-S235114-17 (NET-216-136-220-128-1)	216.136.220.128 - 216.136.220.255
SAVV-S235114-21 (NET-64-56-197-208-1)	64.56.197.208 - 64.56.197.223
LVLT-YAHOO-1-67-72-118 (NET-67-72-118-0-1)	67.72.118.0 - 67.72.119.255

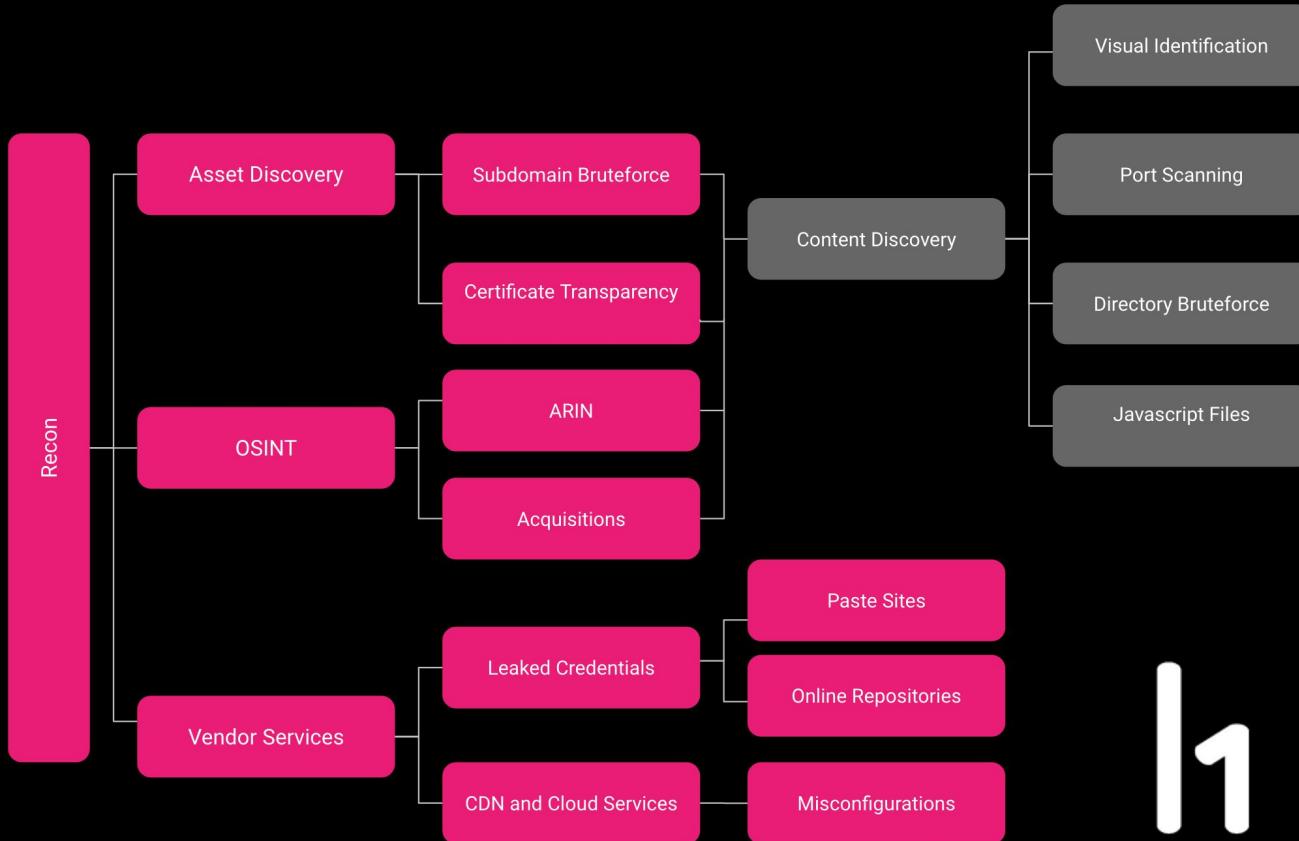
Shodan also helps with this



# Content Discovery

Where the fun begins

# A Visual Guide to Recon



Ben  
Sadeghipour  
(@nahamsec)

# Content Discovery

- Port scan

# Content Discovery

- Port scan
- Screenshot open ports (default: 80, 443)

# Content Discovery

- Port scan
- Screenshot open ports (default: 80, 443)
- Look for interesting
  - Files
  - Directories

# Example

- You see an open port on example.com:8433

# Example

- You see an open port on example.com:8433
- Directory brute force

# Example

- You see an open port on 8433
- Directory brute force
- /admin/ returns 403

# Example

- You see an open port on 8433
- Directory brute force
- /admin/ returns 403
- You brute force for more files/directories on /admin/

# Example

- You see an open port on 8433
- Directory brute force
- /admin/ returns 403
- You brute force for more files/directories on /admin/
- /admin/users.php returns 200

# Example

- You see an open port on 8433
- Directory brute force
- /admin/ returns 403
- You brute force for more files/directories on /admin/
- /admin/users.php returns 200
- Repeat on other domains, ports, folders, etc.

# Content Discovery

- Nmap common ports  
(3868,3366,8443,8080,9443,9091,3000,8000,5900,8081,6000,10000,8181,3306,5000,4000,8888,5432,15672,9999,161,4044,7077,4040,9000,8089,443,7447,7080,8880,8983,5673,7443)
- Take screenshots (`webscreenshot.py`)
- Directory/File brute force
  - dirbuster
  - gograbber
  - gobuster
  - dirsearch
  - Probably more tools out there?

# Content Discovery

- Nmap common ports  
(3868,3366,8443,8080,9443,9091,3000,8000,5900,8081,6000,10000,8181,3306,5000,4000,8888,5432,15672,9999,161,4044,7077,4040,9000,8089,443,7447,7080,8880,8983,5673,7443)
- Take screenshots (`webscreenshot.py`)
- Directory/File brute force
  - Robots.txt sometimes does this for you ¯\\_(ツ)\_/¯
- dirbuster
- gograbber
- gobuster
- dirsearch
- Probably more tools out there?

# Content Discovery

- **ALWAYS** keep an archive of your reports..

# Content Discovery

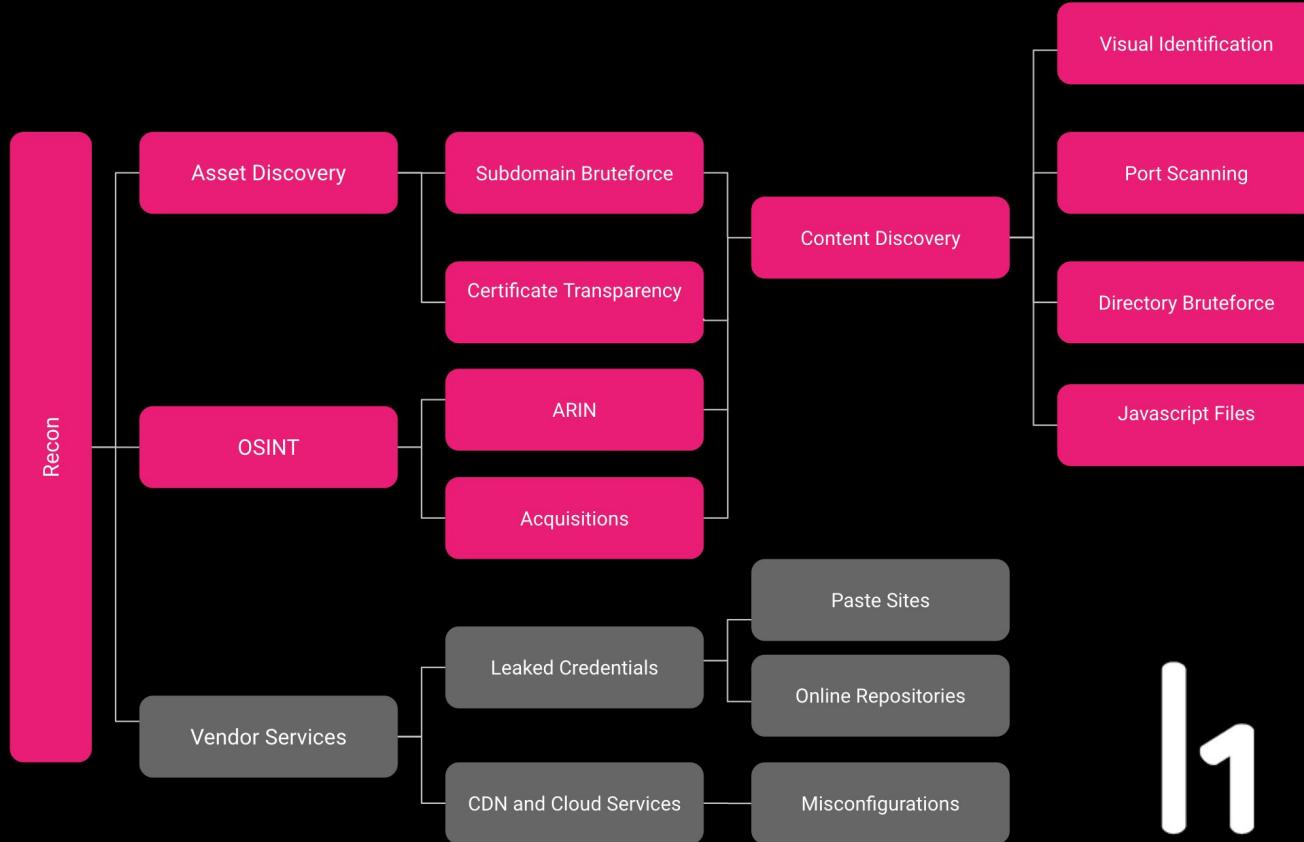
- **ALWAYS** keep an archive of your reports..

```
[root@lazyrecon:~/tools/dirsearch/reports# cat **/* | grep jmx-console | wc -l  
172
```

```
[root@lazyrecon:~/tools/dirsearch/reports# cat **/* | grep manager/html/ | grep 401 | wc -l  
9
```

# Automation

# A Visual Guide to Recon



1

Ben  
Sadeghipour  
(@nahamsec)

# AWS Recon

- Look for S3 buckets on Google  
(site:s3.amazonaws.com +  
inurl:company\_name )
- AWS instances (site:amazonaws.com -s3)
- Repeat on Github!
- Automate your work...

# Australian Broadcasting Corporation leaks passwords, video from AWS S3 bucket

'Advance video content' and years of backups dangled in the cloud

November 02, 2017

Another misconfigured Amazon S3 server leaks data of 50,000 Australian employees



Cyber-Safe

## Pentagon exposed some of its data on Amazon server

# Alteryx S3 leak leaves 123m American

NEWS

## Misconfigured Amazon S3 buckets expose sensitive data

25 SEP 2017 NEWS

### Verizon Hit by Another Amazon S3 Leak



"amazonaws.com" trello

Pull requests Issues Marketplace Gist

Repositories

Code 14K

Commits 28

Issues 213

Wikis 16

Users

Showing 14,361 available code results ⓘ

Sort: Best match ▾



whachoe/whackthedeve – renew\_avatars.sh

Showing the top six matches Last indexed on Sep 16, 2016

Shell

```
1 curl -o jo_giraerts.png https://trell...  
avatars.s3.amazonaws.com/dfc5a54e78c3811891709fca34a6cd73/50.png  
2 curl -o anja_vochten.png https://trell...  
avatars.s3.amazonaws.com/980011062ac6b23313881faf5d68934f/50.png
```



site:s3.amazonaws.com inurl:uber



All News Maps Images Videos More Settings Tools

About 359 results (0.43 seconds)

Drive Now | Uber

devbuilds.uber.com/s3.amazonaws.com/iOS\_UberPartner\_Latest.html ▾

This site uses cookies to provide a personalized and secure experience for users. Cookies allow us to understand user behavior on the site and improve the site.

Dallas uberX – Partner Resources

uber-partners-static.s3.amazonaws.com/dallas\_uberx/index.html ▾

Welcome Uber Dallas Partners and Drivers! Sign Up for uberX · Frequently Asked Questions · Dallas uberX Training. Use the buttons above to sign up, get answers to ...

uberX FAQ Page - Amazon Web Services

uber-partners-static.s3.amazonaws.com/omaha\_uberx/faq.html ▾

When do I get paid? Uber pays partners on a weekly basis from Monday 4:00am to the next Monday at 3:59am. Please allow 3-5 business days for this payment ...

[PDF] Uber Rasier Agreement

https://s3.amazonaws.com/uber.../RASIER%20Technology%20Services%20Agreeme... ▾

Dec 11, 2015 - RASIER, LLC / RASIER-CA, LLC / RASIER-PA, LLC / RASIER-DC, LLC / RASIER-MT, LLC / HINTER-NM. TECHNOLOGY SERVICES ...

[PDF] This study

https://s3.amazonaws.com/uber-static.../Uber\_Driver-Partners\_Hall\_Kreuger\_2015.p... ▾

by J Hall - 2015 - Cited by 162 - Related articles  
Jan 22, 2015 - Dr. Jonathan V. Hall is the Head of Policy Research at Uber Technologies. Prior to joining Uber Technologies in 2014, Dr. Hall held similar ...

[PDF] vehicle inspection - Mr Lucky Auto Service Inc

uber-static.s3.amazonaws.com/la\_dops/uber\_TNC\_inspection\_form\_v9.pdf ▾

1 Foot brakes (pads/shoes thickness). Min. per manufacturer: Front. Rear. Front Brake Left. Measurements. Front Brake Right. Measurements. Rear Brake Left.

# S3 Automation



```
~ — ssh root@45.76.62.224
root@vultr:~/tools/lazys3# [REDACTED]
```

A screenshot of a terminal window titled "ssh root@45.76.62.224". The title bar also includes a "+" button. The terminal shows a single line of text: "root@vultr:~/tools/lazys3# [REDACTED]", where the command line itself is redacted. The background of the terminal window is black.

# S3 Automation

```
root@vultr:~/tools/lazys3# []
```

```
root@vultr:~/tools/teh_s3_bucketeers# ./bucketeer.sh test[]
```

# Certspotter

- Great API
- Easy to automate:
  - Make an alias
  - Automate
  - Win

```
root@vps82152:~# certspotter yahoo.com
170prd.fin.yahoo.com
2013-en-imagenes.es.yahoo.com
360.mail.yahoo.com
3arrebni.yahoo.com
7-eleven.yahoo.com
aa.lrd.yahoo.com
about.yahoo.com
abumediamobile.query.yahoo.com
abumedia.yahoo.com
abumedia.yql.yahoo.com
abuse.corp.yahoo.com
abuse.yahoo.com
academy-delivery.cc.corp.yahoo.com
accmgr.secure.webhosting.yahoo.com
accmgr.webhosting.yahoo.com
accountkey.yahoo.com
accountlink.www.yahoo.com
accountlink.yahoo.com
accountservicer.corp.yahoo.com
ace.ysm.yahoo.com
aclpushdb.ops.yahoo.com
a-cms.shp.corp.tw1.yahoo.com
actapi.corp.yahoo.com
actualites.yahoo.com
adbuilder.creative.yahoo.com
add.my.yahoo.com
address.yahoo.com
adlatencyvendoroutreach.yahoo.com
admanagerplus.yahoo.com
admanager.yahoo.com
admetricsclone.uadapp.yahoo.com
admetrics.uadapp.yahoo.com
adminapp.creatr.corp.yahoo.com
admin.bb.abuse.yahoo.com
admin.bf1.yhs.search.yahoo.com
admin.ckms.yahoo.com
admincms.corp.yahoo.com
admincms.labs.yahoo.com
```

# Create aliases

```
certspotter(){  
    curl -s https://certspotter.com/api/v0/certs?domain=$1 | jq '.[].dns_names[]' | sed 's/^"/g' | sed  
    's/^*\.//g' | sort -u | grep $1 > ~/${1}.txt  
}
```

# Create aliases

```
certspotter(){  
    curl -s https://certspotter.com/api/v0/certs?domain\=$1 | jq '.[].dns_names[]' | sed 's/\\"//g' | sed  
    's/^*\.\//g' | sort -u | grep $1 > ~/${1}.txt  
}  
  
dirbruteforce(){  
    cd /tools/dirsearch  
    cat ~/${1}.txt | while read line; do python3 dirsearch.py -e . -u "https://$line"; done  
}
```

# Create aliases

```
certspotter(){  
    curl -s https://certspotter.com/api/v0/certs?domain=$1 | jq '.[].dns_names[]' | sed 's/\"//g' | sed  
    's/^*\.//g' | sort -u | grep $1 > ~/${1}.txt  
}  
  
dirbruteforce(){  
    cd /tools/dirsearch  
    cat ~/${1}.txt | while read line; do python3 dirsearch.py -e . -u "https://$line"; done  
}  
  
screenshot(){  
    python ~/tools/webscreenshot/webscreenshot.py -o ./${1}/screenshots/ -i ~/${1}.txt --timeout=10 -m  
}
```

# Create aliases

```
certspotter(){  
    curl -s https://certspotter.com/api/v0/certs?domain=$1 | jq '.[].dns_names[]' | sed 's/\"//g' | sed  
    's/^*\.//g' | sort -u | grep $1 > ~/${1}.txt  
}  
  
dirbruteforce(){  
    cd /tools/dirsearch  
    cat ~/${1}.txt | while read line; do python3 dirsearch.py -e . -u "https://$line"; done  
}  
  
screenshot(){  
    python ~/tools/webscreenshot/webscreenshot.py -o ./${1}/screenshots/ -i ~/${1}.txt --timeout=10 -m  
}  
  
recon(){  
    certspotter ${1}  
    dirbruteforce ${1}  
    screenshot ${1}  
}
```

Game changer



# Put your aliases together

```
recon(){  
    certspotter $1  
    dirbruteforce $1  
    screenshot $1  
    [...]  
}
```



## Usage

```
./lazyrecon.sh target.com
```

## About

This script is intended to automate your reconnaissance process in an organized fashion by performing the following:

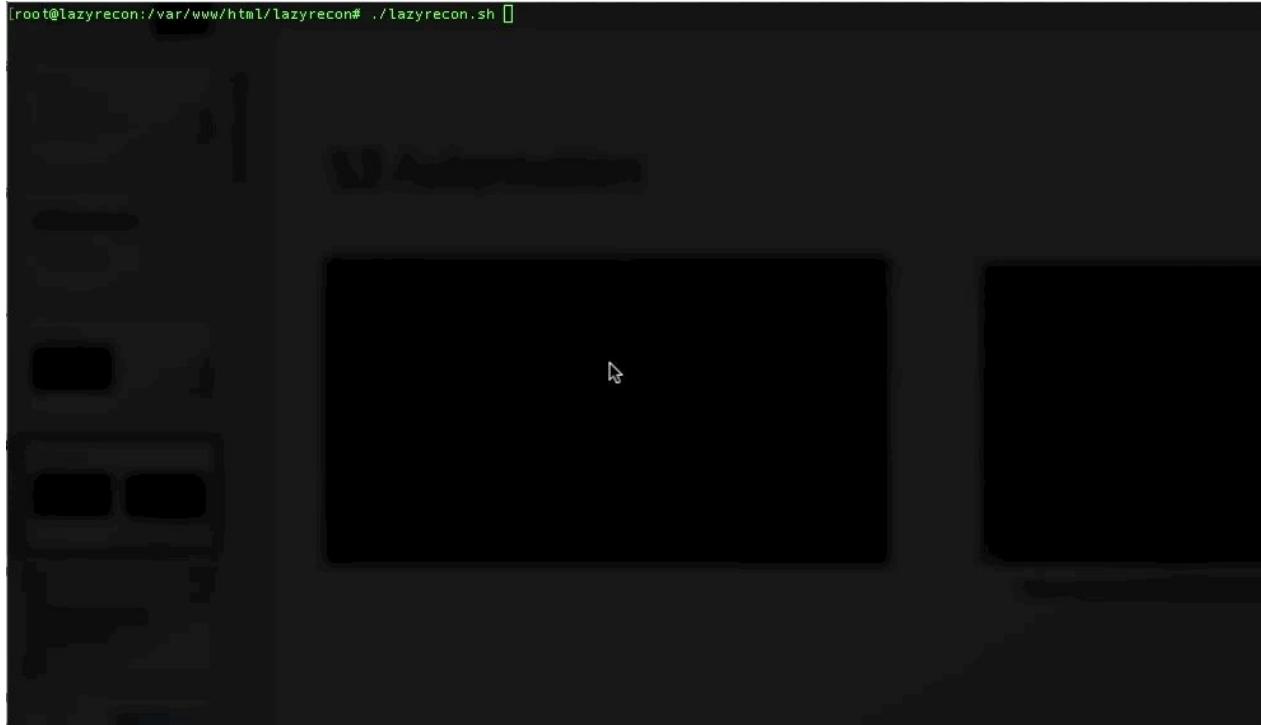
- Create a dated folder with recon notes
- Grab subdomains using Sublist3r and certspotter
- Grab a screenshot of responsive hosts
- Grab the response header
- Perform nmap
- Perform dirsearch
- Generate a HTML report with output from the tools above

This requires Bug Bounty Hunting Tools in order for the tools to work.

**Warning:** This code was originally created for personal use for myself, so it's a bit messy and hopefully it'll be cleaned up with more features in a later release.

# LazyRecon

```
[root@lazyrecon:/var/www/html/lazyrecon# ./lazyrecon.sh ]
```



# LazyRecon

```
chisel-api.trial.nest.com
www.chisel-api.trial.nest.com
energy-partners.trial.nest.com
www.energy-partners.trial.nest.com
home.trial.nest.com
www.home.trial.nest.com
transport.home.trial.nest.com
transport.trial.nest.com
updates.nest.com
utility-api.nest.com
www.utility-api.nest.com
video.nest.com
www.video.nest.com
weather.nest.com
weave-logsink.nest.com
welcome.nest.com
widgets.nest.com
www.widgets.nest.com
ssl.widgets.nest.com
www.ssl.widgets.nest.com
workswith.nest.com
wulfview.nest.com
www.wulfview.nest.com
wn-api.nest.com
www-api-ft.nest.com
www-catalog-api.nest.com
www.wwn-catalog-api.nest.com
wwn5-api.nest.com
ssl.www.nest.com
www.ssl.www.nest.com
www-video.nest.com
aaa-internal.production.nest.com was unreachable
accounts.ft.nest.com is up
accounts.nest.com is up
admin.ft.nest.com is up
admin.ftwest.nest.com was unreachable
admin.home.ft.nest.com was unreachable
admin.home.ftwest.nest.com was unreachable
admin.home.nest.com was unreachable
admin-migration.nest.com is up
```

# LazyRecon

```
chisel-api.trial.nest.com
www.chisel-api.trial.nest.com
energy-partners.trial.nest.com
www.energy-partners.trial.nest.com
home.trial.nest.com
www.home.trial.nest.com
transport.home.trial.nest.com
transport.trial.nest.com
updates.nest.com
utility-api.nest.com
www.utility-api.nest.com
video.nest.com
www.video.nest.com
weather.nest.com
weave-logsink.nest.com
welcome.nest.com
widgets.nest.com
www.widgets.nest.com
ssl.widgets.nest.com
www.ssl.widgets.nest.com
workswith.nest.com
wulfview.nest.com
www.wulfview.nest.com
wn-api.nest.com
www-api-ft.nest.com
www-catalog-api.nest.com
www.wwn-catalog-api.nest.com
wwn5-api.nest.com
ssl.www.nest.com
www.ssl.www.nest.com
www-video.nest.com
aaa-internal.production.nest.com was unreachable
accounts.ft.nest.com is up
accounts.nest.com is up
admin.ft.nest.com is up
admin.ftwest.nest.com was unreachable
admin.home.ft.nest.com was unreachable
admin.home.ftwest.nest.com was unreachable
admin.home.nest.com was unreachable
admin-migration.nest.com is up
```

[+] 73886 URLs to be screenshot



# Recon Report for [home.qa.nestlabs.com](https://home.qa.nestlabs.com)

Generated by LazyRecon on Wed Apr 4 04:42:51 UTC 2018

## Dirsearch

```
401 21B https://home.qa.nestlabs.com:443//users/sign_in
400 166B https://home.qa.nestlabs.com:443%2e%2e//google.com
401 21B https://home.qa.nestlabs.com:443/api-ui
401 21B https://home.qa.nestlabs.com:443/api-docs/
401 21B https://home.qa.nestlabs.com:443/api/docs/
401 21B https://home.qa.nestlabs.com:443/api/
401 21B https://home.qa.nestlabs.com:443/api/error_log
401 21B https://home.qa.nestlabs.com:443/api/
401 21B https://home.qa.nestlabs.com:443/api/build.pyc
200 1KB https://home.qa.nestlabs.com:443/favicon.ico
208 123KB https://home.qa.nestlabs.com:443/login/
208 123KB https://home.qa.nestlabs.com:443/login/administrator/
208 123KB https://home.qa.nestlabs.com:443/login/admin/
208 123KB https://home.qa.nestlabs.com:443/login/cpanel.php
208 123KB https://home.qa.nestlabs.com:443/login/admin/admin.asp
208 123KB https://home.qa.nestlabs.com:443/login/cpanel.zip
208 123KB https://home.qa.nestlabs.com:443/login/cpanel.jsp
208 123KB https://home.qa.nestlabs.com:443/login/cpanel.asp
208 123KB https://home.qa.nestlabs.com:443/login/cpanel.jar
208 123KB https://home.qa.nestlabs.com:443/login/cpanel.sql
208 123KB https://home.qa.nestlabs.com:443/login/cpanel/
208 123KB https://home.qa.nestlabs.com:443/login/index
208 123KB https://home.qa.nestlabs.com:443/login/super
208 123KB https://home.qa.nestlabs.com:443/login/oauth/
208 123KB https://home.qa.nestlabs.com:443/login/cpanel.html
208 123KB https://home.qa.nestlabs.com:443/login/cpanel.aspx
200 123KB https://home.qa.nestlabs.com:443/login/login
200 26B https://home.qa.nestlabs.com:443/robots.txt
401 21B https://home.qa.nestlabs.com:443/sessions/
400 74B https://home.qa.nestlabs.com:443/session
401 21B https://home.qa.nestlabs.com:443/session/
401 21B https://home.qa.nestlabs.com:443/sessions
401 21B https://home.qa.nestlabs.com:443/user.jsp
401 21B https://home.qa.nestlabs.com:443/user.asp
401 21B https://home.qa.nestlabs.com:443/user.php
401 21B https://home.qa.nestlabs.com:443/user.html
401 21B https://home.qa.nestlabs.com:443/user/admin
401 21B https://home.qa.nestlabs.com:443/user/admin.php
401 21B https://home.qa.nestlabs.com:443/user/login.aspx
401 21B https://home.qa.nestlabs.com:443/user.jar
401 21B https://home.qa.nestlabs.com:443/user/
401 21B https://home.qa.nestlabs.com:443/user/login.zip
401 21B https://home.qa.nestlabs.com:443/user/login.jar
401 21B https://home.qa.nestlabs.com:443/user/login.php
401 21B https://home.qa.nestlabs.com:443/user.aspx
401 21B https://home.qa.nestlabs.com:443/user.sgl
401 21B https://home.qa.nestlabs.com:443/user/login.jsp
401 21B https://home.qa.nestlabs.com:443/user.tx
401 21B https://home.qa.nestlabs.com:443/user/
401 21B https://home.qa.nestlabs.com:443/user/loginn.html
```

## Screeshot



## Dig Info

```
; <>> DIG 9.18.3-P4-Ubuntu <>> home.qa.nestlabs.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 39504
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 4, ADDITIONAL: 1
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: udp: 4096
;; QUESTION SECTION:
;home.qa.nestlabs.com. IN A

;; ANSWER SECTION:
home.qa.nestlabs.com. 87 IN CNAME home-hme01-qa-1716690727.us-east-1.elb.amazonaws.com.
home-hme01-qa-1716690727.us-east-1.elb.amazonaws.com. 27 IN A 35.171.205.196
home-hme01-qa-1716690727.us-east-1.elb.amazonaws.com. 27 IN A 34.195.58.159

;; AUTHORITY SECTION:
us-east-1.elb.amazonaws.com. 1183 IN NS ns-1119.awsdns-11.org.
us-east-1.elb.amazonaws.com. 1183 IN NS ns-1793.awsdns-32.co.uk.
us-east-1.elb.amazonaws.com. 1183 IN NS ns-235.awsdns-29.com.
us-east-1.elb.amazonaws.com. 1183 IN NS ns-934.awsdns-52.net.

;; Query time: 0 msec
;; SERVER: 108.61.10.10#53(108.61.10.10)
```

# Digital Dumpster Diving

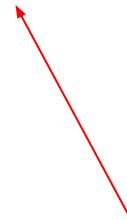
# Digital Dumpster Diving (I'm a pro at it)

# Digital Dumpster Diving

- Leaked credentials
- Leaked api\_tokens
- Leaked authorization headers
- [...]

# Digital Dumpster Diving

- Leaked credentials
- Leaked api\_tokens
- Leaked authorization headers
- [...]



Do you see a pattern here?



# Github Recon

Tools and Keywords

- gitrob
- git-all-secrets
- truffleHog
- git-secrets
- repo-supervisor
- Do it manually?

```
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", "7JyNOrk2387bEr088eg8IfhYjAYdFJlhCbKEo6A");
    props.setProperty("appinfo.validateInstanceId", "false");

    props.setProperty("discovery.us-east-1.availabilityZones", "us-east-1c,us-east-1d,us-east-1e");
```

# Github Recon

Examples

- “company.com” “dev”
- “dev.company.com”
- “company.com” API\_key
- “company.com” password
- “api.company.com” authorization
- GET CREATIVE!

APP\_SECRET  
consumerkey  
JIRA\_Password  
jdbc  
“authorization bearer”  
auth\_key  
consumer\_secret  
SECURITY-SIGNATURE  
X-API  
X-Paypal  
secret\_key  
JWK/JWT  
SSO\_LOGIN  
defaultEndpointsProtocol  
access\_key  
accountKey  
AWS\_Secret  
aws\_secret\_access\_key  
rexis  
api\_key

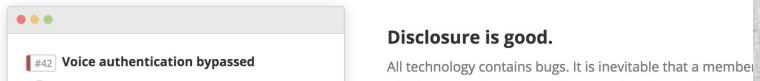
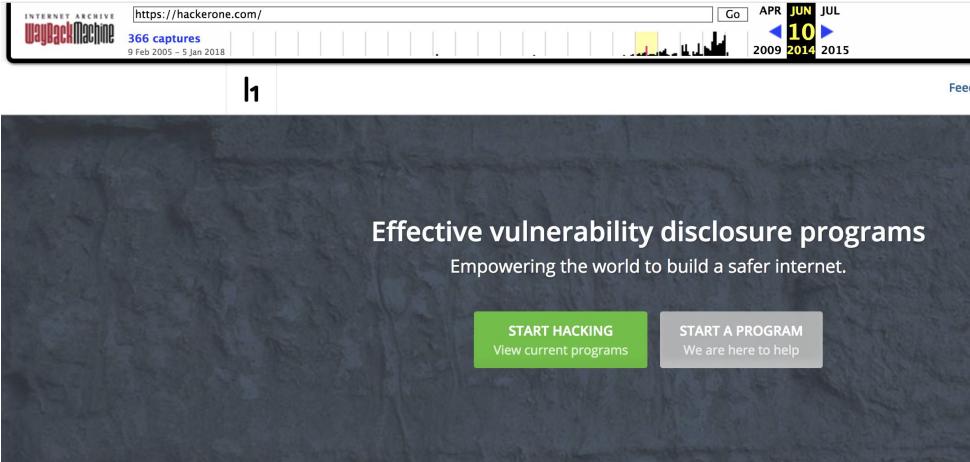
# Archive.org (Wayback Machine)

"That's when I found this. An end of the world party featuring DJ Mobley. I'm not an EDM fan myself, but a quick search on the **wayback** machine, and guess what we found all the way from 2003. **You.** An Angelfire page that you created from back in the day. Your very own DJ Mobley fan page."



# Archive.org

- Search for your target
- Select a date
- Review the source -> find JS files
- Find old endpoints/functionality
- Exploit them!



# Javascript Files

Why?

# Javascript Files

- Look for (hidden) endpoints
- Leaked cloud instances and secret\_keys
- ... and definitely more bugs

# Javascript Files

- Look for (hidden) endpoints
- Leaked cloud instances and their secret\_keys
- ... and definitely more bugs

## JSParser

A python 2.7 script using Tornado and JSBeautifier to parse relative URLs from JavaScript files. Useful for easily discovering AJAX requests when performing security research or bug bounty hunting.

## Dependencies

- safeurl
- tornado
- jsbeautifier

## Installing

```
$ python setup.py install
```

## Running

Run `handler.py` and then visit <http://localhost:8008>.

```
$ python handler.py
```

# Javascript Files

Examples

- Look for (hidden) endpoints
- Leaked cloud instances and their secret\_keys
- ... and definitely more bugs

JS Parser Home

```
/v1/help/submit_contact
2669: return e.save_contact_us_only = 10, !isEmpty(e.message) && (e.message = "Created for Matchbox"), $.post(R.default.getUrl("/v1
/help/submit_contact"), e).then(function(e) {
```

```
/v1/help/issues
3086: var i = "/v1/help/issues/" + String(e),
```

```
/v2/channels
3700: return r.default.get("/v2/channels", {
```

```
/chat
3724: babelHelpers.classCallCheck(this, e), this.baseUrl = t.baseUrl || "/chat"
```

```
/availability
3730: return r.default.getJSON(String(this.baseUrl) + "/availability", {
```

```
/estimatedWaitTime
3740: return r.default.getJSON(String(this.baseUrl) + "/estimatedWaitTime", {
```

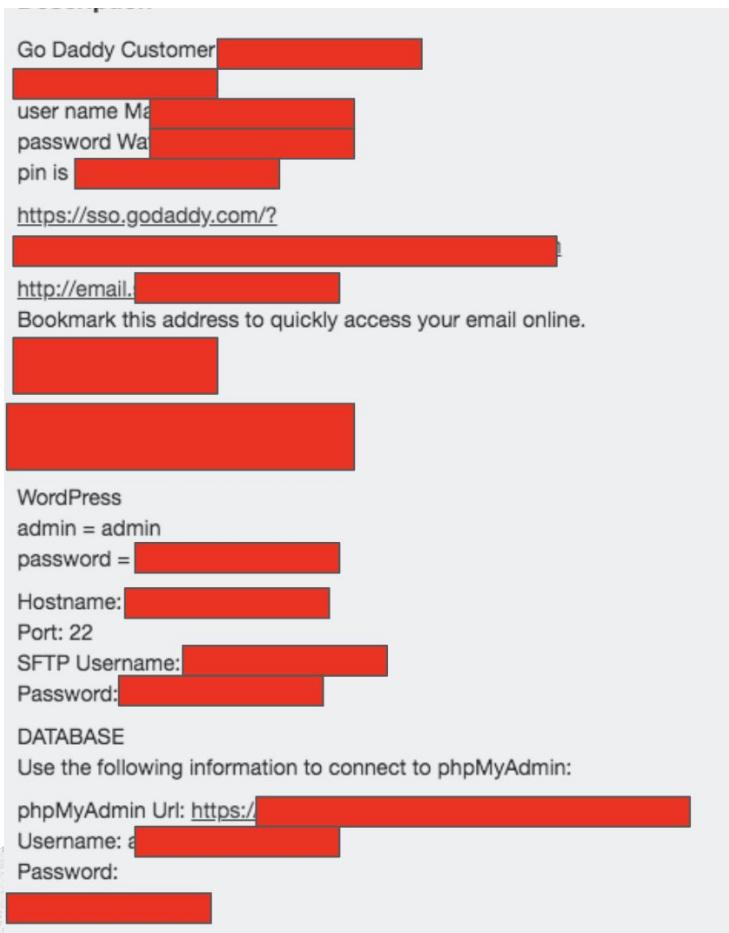
```
/request
3772: url: String(this.baseUrl) + "/request",
```

```
/events
3799: url: String(this.baseUrl) + "/" + String(t) + "/events",
```

```
/info/visitorTyping
3837: url: String(this.baseUrl) + "/" + String(t) + "/info/visitorTyping",
```

# Trello Boards

- Site:trello.com intext:ftp
- Site:trello.com intext:ORG



# Trello Boards

- Site:trello.com intext:ftp
- Site:trello.com intext:ORG



Go Daddy Customer [REDACTED]

[REDACTED]  
user name Ma [REDACTED]  
password Wa [REDACTED]  
pin is [REDACTED]

[https://sso.godaddy.com/?](https://sso.godaddy.com/)

[REDACTED]  
[http://email.\[REDACTED\]](http://email.[REDACTED])

Bookmark this address to quickly access your email online.

[REDACTED]  
[REDACTED]

WordPress  
admin = admin  
password = [REDACTED]

Hostname: [REDACTED]  
Port: 22

SFTP Username: [REDACTED]  
Password: [REDACTED]

DATABASE

Use the following information to connect to phpMyAdmin:

phpMyAdmin Url: [https://\[REDACTED\]](https://[REDACTED])  
Username: a [REDACTED]  
Password: [REDACTED]

[REDACTED]

# Examples

# Examples of Certificate Transparency

# Shodan Examples

Search Query: hostname:host.com port:15672

---

  #100928 Access to RabbitMQ on stageREDACTED.REDACTED.com:15672

Search Query: hostname:host.com title:Dashboard [Jenkins]

---

  #220835 jenkins-REDACTED.REDACTED.REDACTED.com publicly facing without authentication leaks AWS\_Secret\_key + Build info

# Censys Examples

- Working example:



## Script Console

Type in an arbitrary [Groovy script](#) and execute it on the server. Useful for trouble-shooting and diagnostics. Use the 'println' command to see the output (if you use `System.out`, it will go to the server's stdout, which is harder to see.) Example:

```
println(Jenkins.instance.pluginManager.plugins)
```

All the classes from all the plugins are visible. `jenkins.*`, `jenkins.model.*`, `hudson.*`, and `hudson.model.*` are pre-imported.

```
1 def sout = new StringBuilder(), serr = new StringBuilder()
2 def proc = 'echo testing for RCE :'.execute()
3 proc.consumeProcessOutput(sout, serr)
4 proc.waitForOrKill(1000)
5 println "RCE> $sout err> $serr"
```

Run

### Result

```
RCE> testing for RCE :)
err>
```

# Examples of Discovering Endpoints Hidden Inside of Javascript Files

# Airbnb – Web to App Phone Notification IDOR to view Everyone's Airbnb Messages

```
"/v2/air_push_notifications
```

```
7629: return "/v2/air_push_notifications";
```

```
"/v2/air_sms_notifications
```

```
7631: return "/v2/air_sms_notifications";
```

# Airbnb – Web to App Phone Notification IDOR to view Everyone's Airbnb Messages

## API Requests

[https://www.airbnb.com/api/v2/air\\_sms\\_notifications](https://www.airbnb.com/api/v2/air_sms_notifications)

- Requires that you have a verified phone number in your profile.
- This API request allowed you to send yourself SMS texts.
- There is a length limit on the SMS messages (160)
- Throttling restrictions on SMS only allowed you to send this API request a handful of times every hour

# Airbnb – Web to App Phone Notification IDOR to view Everyone's Airbnb Messages

## API Requests

[https://www.airbnb.com/api/v2/air\\_push\\_notifications](https://www.airbnb.com/api/v2/air_push_notifications)

- Requires that you have a verified phone number, the Airbnb app installed on your phone (with that phone number), and you are logged into the app with that account. (This one took awhile to figure out)
- Instead of sending SMS, it pushed notifications to your phone through the phone app.
- There is no length restriction on the output.
- No throttling which made testing a lot easier compared to SMS

# Airbnb – Web to App Phone Notification IDOR to view Everyone's Airbnb Messages

## POST REQUEST:

```
{"_format":"for_visitor","country":"USA","phone_number":"","template":"message","user_id":,"title":"","body":"","metadata":{},"object_id":"","status":"","role":"","photo_url":""}
```

If you passed in an invalid template it would give you a list of all the valid templates that you could send.

## Templates

/api/v2/air\_sms\_notifications

content\_framework, custom,  
host\_banner\_app\_install,  
host\_never\_actives\_just\_raw\_listing,  
host\_never\_actives\_on\_description,  
host\_never\_actives\_on\_photo,  
host\_never\_actives\_on\_price\_or\_booking\_setting,  
message, mobile\_photo\_upload\_app\_install,  
identity\_verifications,  
identity\_verifications\_booking,  
p2\_p3\_abandon\_sms,  
reservation\_alteration\_approved,  
reservation\_alteration\_declined,  
reservation\_guest\_accepted,  
reservation\_guest\_cancelled,  
reservation\_guest\_declined,  
reservation\_host\_accepted,  
reservation\_host\_cancelled,  
reservation\_host\_declined,  
verified\_id\_app\_install,  
review\_finalReminder\_message,  
mt\_pdp\_handoff, mt\_native\_handoff\_generic,  
message\_image\_attachment,  
guidebook\_landing\_page,  
wish\_lists\_native\_handoff

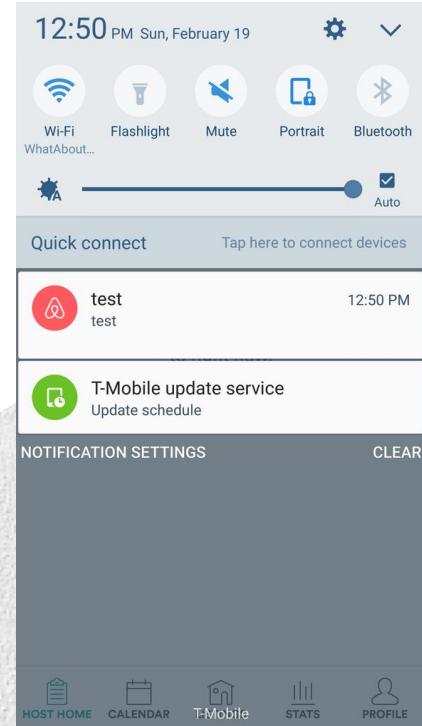
/api/v2/air\_push\_notifications

checkpoint, cn\_blackout\_g20\_hangzhou\_2016, custom,  
host\_never\_actives\_just\_raw\_listing,  
host\_never\_actives\_on\_description,  
host\_never\_actives\_on\_photo,  
host\_never\_actives\_on\_price\_or\_booking\_setting,  
identity\_verifications, identity\_verifications\_booking,  
message\_image\_attachment, message,  
midtrip\_host\_checkup\_reminder, mobile\_photo\_upload,  
paid\_amenity\_accepted,  
paid\_amenity\_canceled\_by\_guest,  
paid\_amenity\_canceled\_by\_host, paid\_amenity\_declined,  
paid\_amenity\_request, paid\_amenity\_shop\_services,  
preapproval\_guest\_sent, preapproval\_guest\_withdrawn,  
reservation\_alteration\_approved,  
reservation\_alteration\_request\_automatically\_accepted,  
reservation\_alteration\_declined,  
reservation\_alteration\_request,  
reservation\_guest\_accepted, reservation\_guest\_cancelled,  
reservation\_guest\_declined, reservation\_host\_accepted,  
reservation\_host\_cancelled, reservation\_host\_declined,  
reservation\_host\_request, reservation\_payment\_pending,  
reservation\_host\_first\_reminder,  
reservation\_host\_last\_reminder,  
review\_finalReminder\_message, risk\_email\_updated,  
risk\_password\_updated, risk\_payout\_method\_updated,  
risk\_phone\_number\_updated, share\_your\_trip\_prompt,  
special\_offer\_guest, special\_offer\_guest\_expired,  
special\_offer\_guest\_withdrawn,  
special\_offer\_host\_expired, support\_message

# Airbnb – Web to App Phone Notification IDOR to view Everyone's Airbnb Messages

Then we found out we can send ourselves custom messages:

```
{"_format": "for_visitor", "country": "USA", "phone_number": "", "template": "custom", "user_id": 109764261, "status": "test", "title": "test", "body": "test"}
```



# Airbnb – Web to App Phone Notification IDOR to view Everyone's Airbnb Messages

Then we found out we can send ourselves custom messages:

```
{"_format": "for_visitor", "country": "USA", "phone_number": "", "template": "custom", "user_id": 109764261, "status": "test", "title": "test", "body": "test"}
```

**What else can we do?**

# Airbnb – Web to App Phone Notification IDOR to view Everyone's Airbnb Messages

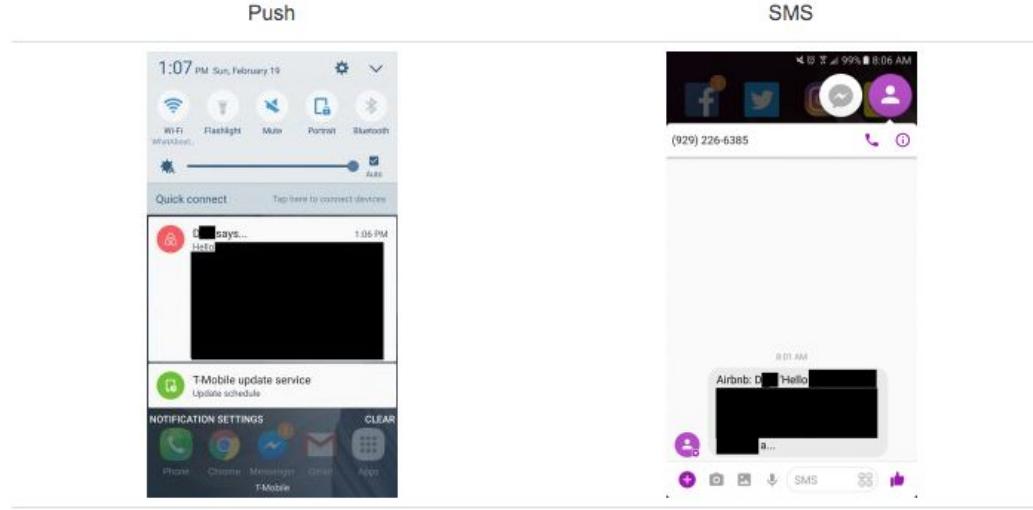
Then we found out we can send ourselves custom messages:

```
{"_format": "for_visitor", "country": "USA", "phone_number": "", "template": "custom", "user_id": 109764261, "status": "test", "title": "test", "body": "test"}
```

## What else can we do?

Perhaps enumerate on user\_id or object\_id in each request and get access to other users' notifications?

# Airbnb – Web to App Phone Notification IDOR to view Everyone's Airbnb Messages



# Examples of Digital Dumpster Diving

# Dumpster Diving Part 1

- Looked up the “umbrella” company name

# Dumpster Diving Part 1

- Looked up the “umbrella” company name
- Combine “umbrella\_company” + asset\_name + “password”, and found this code on Github:

```
"server": {  
    "host": "dedXXXX.PATTERN.PROVIDER.com",  
    "port": 21,  
    "user": "some_username",  
    "password": "definitely_ftp_passwords"  
}
```

# Dumpster Diving Part 1

- Looked up the “umbrella” company name
- Combine “umbrella\_company” + asset\_name + “password”, and found below code:

```
"server": {  
    "host": "dedXXXX.PATTERN.PROVIDER.com",  
    "port": 21,  
    "user": "some_username",  
    "password": "definitely_ftp_passwords"  
}
```

- Got access to umbrella\_company’s FTP server



Best\_company\_ever rewarded nahamsec with a \$10,000 bounty.

Dec 28th (8 months ago)

# Dumpster Diving Part 2

#343509

## Multiple FTP and internal credentials leaked on Github

State • Resolved (Closed)

Severity Medium (4 ~ 6.9)

Reported To Some Company

Participants  (Manage collaborators)

Weakness Cleartext Storage of Sensitive  
Information

Visibility Private

Bounty \$750

# Dumpster Diving Part 2

#343509    Multiple FTP and internal credentials leaked on Github

---

#345154    Leaked AWS instance keys + database credentials on github leads to database access and possible RCE

Weakness	State	Resolved (Closed)	Severity	Critical (9 ~ 10)
	Reported To		Participants	
	Asset		Visibility	Private
Weakness	Improper Access Control - Generic			
Bounty	\$450			

# Dumpster Diving Part 2

#343509    Multiple FTP and internal credentials leaked on Github

---

Repo: #345154 Leaked AWS instance keys + database credentials on github leads to database access and possible RCE

Weakness: #296038 Multi Paypal username, password, signatures leaked on github allows to access your account via the API

Weakness	State	Resolved (Closed)	Severity	No Rating (---)
Bounty	Reported To		Participants	(Manage collaborators)
Weakness	Plaintext Storage of a Password	Visibility	Private	
Bounty	\$500			

# Dumpster Diving Part 2

#343509    Multiple FTP and internal credentials leaked on Github

Repo: #345154    Leaked AWS instance keys + database credentials on github leads to database access and possible RCE

Weakness: #296038    Multipl Paypal username, password, signatures leaked on github allo ws to access your account via the API

Weakness: #215500    Leaked FTP credentials on github leads to RCE on amex.someothersit e.com

State: Resolved (Closed)    Severity: No Rating (---)

Bounty: \$1,000    Participants: (Manage collaborators)

Weakness: Command Injection - Generic    Visibility: Private

Bounty: \$1,000

[Collapse](#)

# Example of Readable/Writable S3 Buckets

# AWS CLI

- Requires AWS CLI
- \$ aws s3 ls s3://bucket-name
- \$ aws s3 cp hax0r.txt s3://bucket-name

Pete (yaworsk) 5623 63rd 5.79 93rd 17.89 89th  
Reputation Rank Signal Percentile Impact Percentile

#128088 AWS S3 bucket writeable for authenticated aws users Share: [f](#) [t](#) [g](#) [in](#) [v](#) [e](#)

13 State: Resolved (Closed)  
Disclosed publicly: April 5, 2016 6:06am -0700 Severity: No Rating (---)  
Reported To: HackerOne Participants:   
Weakness: Improper Authentication - Generic Visibility: Public (Full)  
Bounty: \$2,500 [Collapse](#)

SUMMARY BY HACKERONE  
**h** An ACL misconfiguration issue existed on one of our S3 buckets. This misconfiguration allowed any authenticated AWS user to write to this bucket (no read access was permitted). An attacker could theoretically post a file into that bucket that may at some point be accessed by a HackerOne staff member, thinking it's been uploaded by another staff member or some automated system. We improved the ACLs for that S3 bucket to prevent such a concern.  
This issue also led us to audit some of our additional S3 buckets, resulting in changes for some of those buckets as well.

SUMMARY BY YAWORSK  
**yaworsk** For those interested, I've recorded a video tutorial on how I accomplished this: [https://www.youtube.com/watch?v=\\_x5VKuFjvrk](https://www.youtube.com/watch?v=_x5VKuFjvrk)

TIMELINE  
**yaworsk** submitted a report to **HackerOne**. Apr 3rd (2 years ago)  
**yaworsk** Hi All,  
I know that hackerone-attachments is used for file uploads on reports and so I did a quick scan for similar buckets and found [REDACTED]. While I can't confirm if you own it or not, it appears that it is publicly writable using the aws cli.  
When I tried to write to hackerone-attachments, I get:  
"move failed: ./test.txt to s3://hackerone-attachments/test.txt A client error (AccessDenied) occurred when calling the PutObject operation: Access Denied.  
However, when I write to [REDACTED], I get:  
move: ./test.txt to s3://[REDACTED]/test.txt  
Hopefully the bucket is yours and this isn't a waste of time. If you do own it, a good thing is the bucket is not publicly readable and the file appears private by default after being written. However, assuming you own it, the security issue would be someone writing something malicious and someone on your team unknowingly opening it.

Pete



# CNAME Pointing to Unclaimed AWS S3



Danil Gribkov (dpgrbikov)

350

-

Reputation

2.90

78th

Signal

Percentile

22.50

95th

Impact

Percentile



96

#186766

Subdomain takeover on [happymondays.starbucks.com](#) due to non-used AWS S3 DNS record

Share:



State ● Resolved (Closed)

Severity  High (7 ~ 8.9)

Disclosed publicly December 19, 2016 2:59pm -0800

Participants

Reported To Starbucks

Visibility Public (Full)

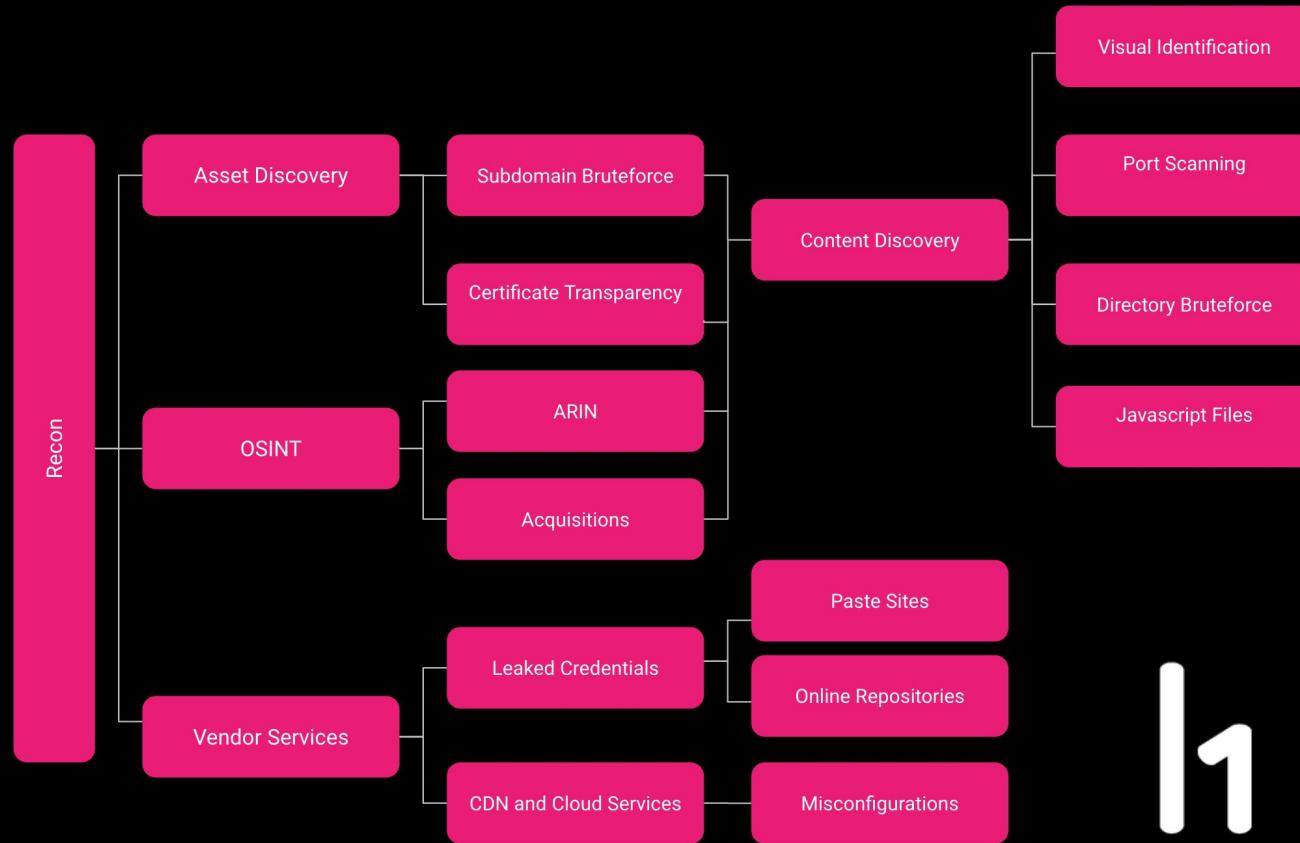
Weakness Privilege Escalation

Bounty \$2,000

[Collapse](#)

Recon Automation +  
Github +  
Exploitation

# A Visual Guide to Recon



Ben  
Sadeghipour  
(@nahamsec)

# Access to all internal API endpoints?

- Asset discovery <https://stgsomethjngsomething.something.target.com/>
- Content discovery:  
<https://stgsomethjngsomething.something.target.com/internal> and

# Access to all internal API endpoints?

- Asset discovery <https://stgsomethingsthingsomething.something.target.com/>
  - Content discovery:  
<https://stgsomethingsthingsomething.something.target.com/internal> and
  - Github searches:

```
resp=$(curl -i -s -S "http://$IP:$PORT/internal/login" -X POST \
-H "Content-Type: application/json" \
-H 'Accept: application/json, text/javascript, */*; q=0.01' \
-H 'Connection: keep-alive' -d '{"username":"'${username}'","password":"'${password}'"}')
COOKIE=$(echo "$resp" | grep set-cookie | awk '{print $4}' ;)
```

# Access to all internal API endpoints?

- Asset discovery <https://stgsomethjngsomething.something.target.com/>
- Content discovery:  
<https://stgsomethjngsomething1.something.target.com/internal> and
- Github searches:

```
resp=$(curl -i -s -S "https://[REDACTED].target.com/internal/login" -X POST \  
    -H "Content-Type: application/json" \  
    -H 'Accept: application/json, text/javascript, */*; q=0.01' \  
    -H 'Connection: keep-alive' -d '{"username":"[REDACTED]","password":"[REDACTED]"}')  
COOKIE=$(echo "$resp" | grep set-cookie | awk '{\$1=""}; print \$0}') ;\
```

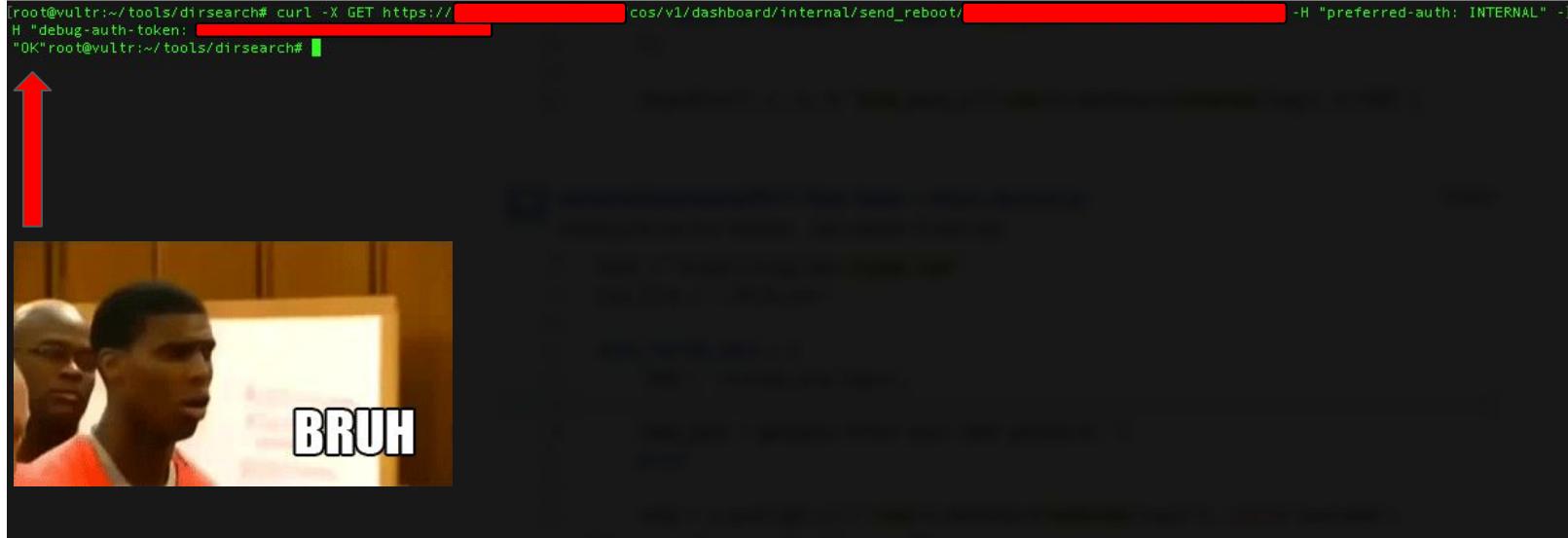
- Looked for the: hostname+ internal + “auth-token” (on github and found this handy curl command:

# Access to all internal API endpoints?

```
root@vultr:~/tools/dirsearch# curl -X GET https://[REDACTED]/v1/dashboard/internal/send_reboot/[REDACTED] -H "Authorization: [REDACTED]-auth: INTERNAL" -d "OK"root@vultr:~/tools/dirsearch#
```

# Access to all internal API endpoints?

```
[root@vultr:~/tools/dirsearch# curl -X GET https://[REDACTED]cos/v1/dashboard/internal/send_reboot/[REDACTED] -H "preferred-auth: INTERNAL" -H "debug-auth-token: [REDACTED]" "OK"root@vultr:~/tools/dirsearch#
```



A red arrow points upwards from the terminal window to a video frame showing two men. One man in an orange shirt is shouting "BRUH".

# Access to all internal API endpoints?

- Looked into the JS files on the login page
- Found all of the internal API calls
- Already have an AUTH-TOKEN that works without login
- Combine all of the above:

```
curl -X GET  
"https://devsomething.something.target.com/ internal/internal_somet  
hing/internal_something_accounts" -H "SOME_KEYWORD-auth: INTERNAL"  
-H "debug-token: SORRY_I_HAD_TO_REDACT"
```

# Access to all internal API endpoints?

- Looked into the JS files on the login page
- Found all of the internal API calls
- Already have an AUTH-TOKEN that works without login
- Combine all of the above:

```
curl -X GET  
"https://devsomething.something.target.com/ internal/internal_accounts" -H "SOME_KEYWORD-auth: INTERNAL" -H "debug-token:  
SORRY_I_HAD_TO_REDACT"
```

Return all LDAP usernames for brute force ;)?

# FINISH HIM !!



rewarded nahamsec with a \$4,348 bounty and a \$192 bonus.

# Keep in touch

- **@nahamsec** on all social media
- Check out all of our programs on [hackerone.com/directory](https://hackerone.com/directory)
- [Bensdp@hackerone.com](mailto:Bensdp@hackerone.com) | [im@ha.cker.af](mailto:im@ha.cker.af)

# Tools

- **Dirsearch** - <https://github.com/maurosoria/dirsearch>
- **JSParser** - <http://github.com/nahamsec/jsparser>
- **LazyS3** - <https://github.com/nahamsec/lazys3>
- **LazyRecon** - <https://github.com/nahamsec/lazyrecon>
- **Teh\_s3\_bucketeers** - [https://github.com/tomdev/teh\\_s3\\_bucketeers](https://github.com/tomdev/teh_s3_bucketeers)

# Thank you

- **Thank YOU** for letting me present
- The **Hacker Community** for being so welcoming and sharing their ideas
- A big **thank you** to: Ziot, Tomdev, ITSecurityguard, Jon Bottarini, Luke Tucker, Smiegles, Jobert Abma, Michiel Prins.

# Thank You