

FAST RECON GG

[Reconocimiento rápido]

[Y LDF en el programa de Google VRP]

by `omespino@google`ESCAL8:~/london\$

whoami

```
omespino@durivacon:~/cdmx$ id
```

Omar Espino aka @omespino [M É X I C O]

por las mañanas:

- infrastructure & security director at SRAX
- background: unix* lover, backend & mobile developer

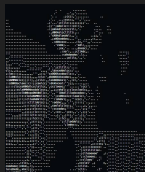
por las noches:

- investigador de seguridad informática independiente
- cazarrecompensas (bug hunter)

blog personal:

<https://omespino.com>

twitter: @omespino



```
omespino@durivacon:~/cdmx$ cat hitos.txt
```

fuí publicado en el blog de chema alonso "*un informático del lado del mal*"



reconocido por / salón de la fama de seguridad informática:



Microsoft



slack

NOKIA



ATLASSIAN

YAHOO!



SONY



libro: Bug bounty en español amazon

fast recon GG (from global to granular) by [omespino@googleESCAL8:~/london\\$](#)

agenda

```
omespino@googleESCAL8:~/london$ cat agenda.txt
```

- introduction
- main bug bounty recon flow
 - global organization discovery
 - subdomain discovery
 - visual identification
 - assets brute forcing / web scraping
- LFI on springboard.google.com recap
- what's nexts? duck test
- lessons learned
- Q&A

WARNING!

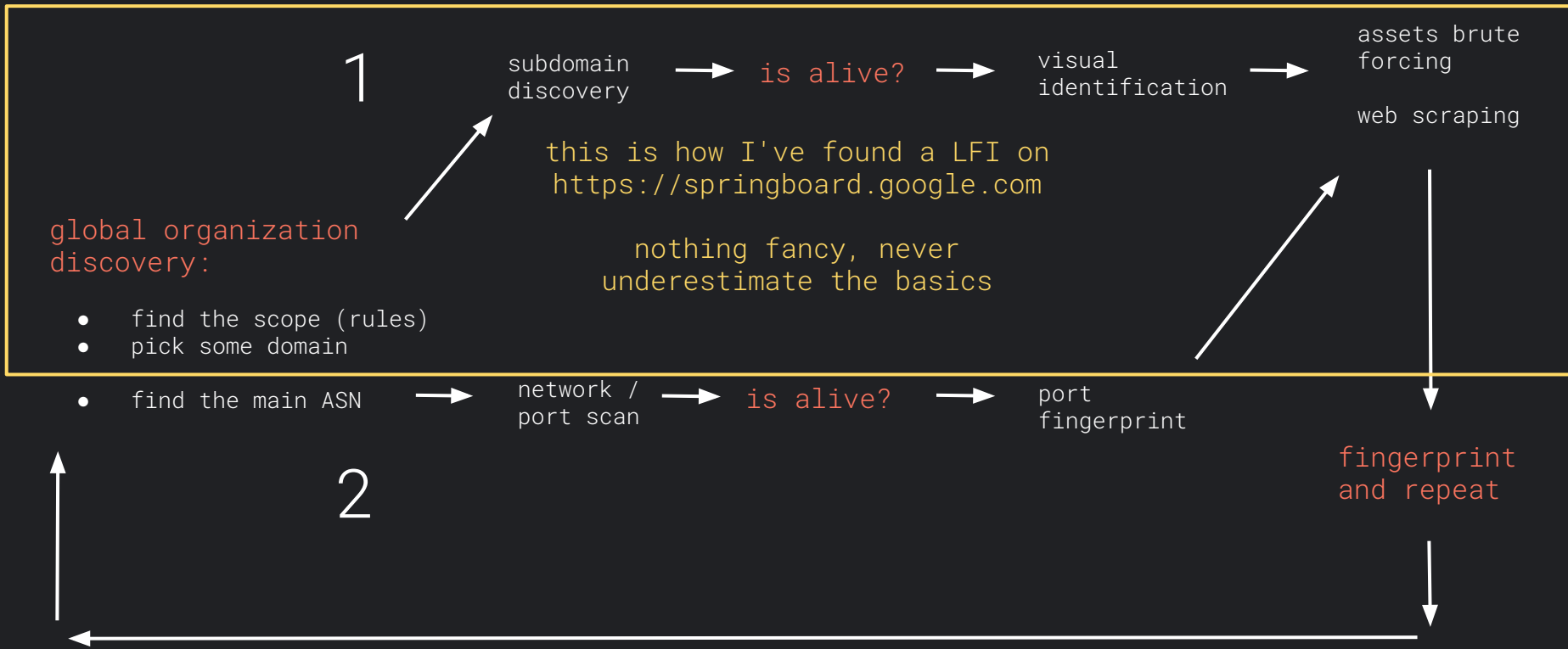


All information included in this medium is for educational and professional purposes, in no case GOOGLE or neither I, are responsible for any misuse of this information.

introduction

[motivation: challenge, fun and profit]

main bug bounty recon flow - keep it simple



global organization discovery

Google Vulnerability Reward Program scope
\$100 - \$31,337 [[g.co/vrp](https://www.google.com/vrp/)]

- *.google.com
- *.youtube.com
- *.blogger.com

(+20,000 subdomains?)

just the main ASN Google LLC
~9 million ip addresses

where should we start?
let's pick *.google.com

1. subdomain discovery - [*].google.com]

aggressive

- to save time choose one of most common environments + tld and start doing some testing: [corp, dev, development, admin, staggin, alpha, stage, prod, beta, local, test].

environment + top level domain
corp.google.com

- then brute force with subbrute.py, massdns & all.txt - thx @jhaddix

```
"/subbrute.py all.txt corp.google.com |  
massdns -r resolvers.txt -t A -a -o -w gcorp.txt -"
```

Total requests - 2,286,549	another tools like
Realtime: - 2m48s !!!	dns-parallel-prober or
	gobuster took about 30m

**all.txt is a combination of wordlists from every public dns source / enumeration tool*

passive

- shodan & zoomeye:
"ssl:corp.google.com"
"hostname:corp.google.com"
- zoomeye, binaryedge.io & github
"corp.google.com"
- censys
"443.https.tls.certificate.parsed.extensions.subject_alt_name.dns_names:corp.google.com" - thx @nahamsec
- crt.sh
"%corp.google.com"
- lifehack - "hack" hackers hacking their hacks. TLDR; grab bug bounty scans results in github lol
thx @randomrobbie

Finally combine all results in 1 file
corp.google.com-all.txt

fast recon GG (from global to granular) by omespino@googleESCAL8:~/london\$

2. network / port scan - [main ASN Google LLC (AS15169)] - NOT IN THE CURRENT FLOW TO FIND THE LFI - BONUS

*extracted from <https://ipinfo.io/AS15169> (9 Million hosts), there are more ASN that belong to google

aggressive

Getting corp.google.com's CIDR

```
whois `dig +short corp.google.com | tail -n1` | grep CIDR  
64.233.160.0/19
```

nmap scan

```
time nmap -p80 -T5 -Pn --max-rate 10000 64.233.160.0/19  
Hosts:      - 2,591  
Realtime:   - 2m 8s
```

masscan scan

```
time masscan --rate 10000 -p80 64.233.160.0/19  
Hosts:      - 2,611  
Realtime:   - 0m 12s
```

masscan > nmap (*SPEED UP 10.6x w00t!*)
BUT, WAIT A SECOND ...

if masscan is faster and nmap is very good at
fingerprinting, first filter alive host with masscan,
then re-scan with nmap

masscan + nmap = FAST AND PRECISE

passive

• shodan:

```
"net:64.233.160.0/19"  
"asn:AS15169"  
"ssl:corp.google.com"
```

• binaryedge.io

```
"ip:64.233.160.0/19"  
"asn:AS15169"  
"ssl:corp.google.com"
```

• zoomeye

```
"cidr:64.233.160.0/19"  
"asn:AS15169"  
"ssl:corp.google.com"
```

• censys

```
"64.233.160.0/19"  
autonomous_system.description.raw: "GOOGLE -  
Google LLC"
```

information already
sorted for you

- top ports
- top services
- top OS
- CVE info
- locations

visual identification

httprobe - @tomnomnom

Take a list of domains and probe for working http and https servers.

You can set the concurrency level with the -c flag.

```
time cat corp.google.com-all.txt | httprobe -c 1000
```

EyeWitness - @ChrisTruncer

EyeWitness is designed to take *screenshots* of websites provide some *server header info*, and *identify default credentials* if known.

The --timeout flag is completely optional, and lets you provide the max time to wait when trying to render and screenshot a web page, *by default timeout is 7 seconds*

```
python ./EyeWitness.py -f ./corp.google.com-probed.txt  
--headless -d corp.google.com-output --timeout 1
```

corp.google.com-all.txt
(~12,312 hosts)

httprobe

EyeWitness - screenshots / headers

subdomain
discovery



are this
hosts alive?



visual
identification

massdns brute forcing,
shodan, zoomeye, censys,
crt.sh, github @randomrobbie
and another hackers hacks
lol

corp.google.com-probed.txt
in ~12,312 hosts out 1023 hosts

screenshots, headers and html
source from 1023 alive hosts

fast recon GG (from global to granular) by omespino@googleESCAL8:~/london\$

visual identification - Eyewitness

```
cd corp.google.com-probed; python -m SimpleHTTPServer 9090 #(9090 or any port)
```

Table of Contents	
	<ul style="list-style-type: none">• Uncategorized (Page 1)• 401/403 Unauthorized (Page 40)• 404 Not Found (Page 40)• Bad Request (Page 40)
Uncategorized	976
401/403 Unauthorized	4
404 Not Found	11
Bad Request	4
Errors	19
Total	1014

Report Generated on XX/XX/2019 at 19:40:39
[Next Page](#)

[Page 1](#) [Page 2](#) [Page 3](#) [Page 4](#) [Page 5](#) [Page 6](#) [Page 7](#) [Page 8](#) [Page 9](#) [Page 10](#) [Page 11](#) [Page 12](#) [Page 13](#) [Page 14](#) [Page 15](#) [Page 16](#) [Page 17](#) [Page 18](#) [Page 19](#) [Page 20](#) [Page 21](#) [Page 22](#) [Page 23](#) [Page 24](#) [Page 25](#) [Page 26](#) [Page 27](#) [Page 28](#) [Page 29](#) [Page 30](#) [Page 31](#) [Page 32](#) [Page 33](#) [Page 34](#) [Page 35](#) [Page 36](#) [Page 37](#) [Page 38](#) [Page 39](#) [Page 40](#) [Page 41](#)

<http://springboard.corp.google.com>
Resolved to: 74.125.197.129

Page Title: Google Cloud Search
strict-transport-security: max-age=31536000
x-content-type-options: nosniff
content-security-policy: script-src 'report-sample' 'nonce-wQmlypyjOP5t1OQh+ZPJhQ' 'unsafe-inline';object-src 'none';base-uri 'self';report-uri /_/TopazUi/cspreport;worker-src 'self'
set-cookie:
NID=189=pcat0kXawynR2EE84yoHkn7l30WLRyO05CsfBOsaHZGTkt9Tphdb__1UH33AFG_q0ygMBfTm7ldvn4sEMiEfUz79H1RGiOs2Xlk1-mkPtXumfACoUc1XrSOp21MSMlyB765zQgilnBK6m8oQOhHxm7mchCglcFk2PPhP1uZSYs; expires=Mon, 13-Apr-2020 20:02:09 GMT; path=/; domain=.google.com; HttpOnly
accept-ranges: none
expires: Mon, 01 Jan 1990 00:00:00 GMT
vary: Accept-Encoding
server: ESF
connection: close
x-xss-protection: 0
x-ua-compatible: IE=edge
pragma: no-cache
cache-control: no-cache, no-store, max-age=0, must-revalidate



Work accounts only

Google Cloud Search is only available for work accounts.

Have another work account?

[SWITCH ACCOUNT](#)

I just `curl -IL https://springboard.corp.google.com`
and `springboard.corp.google.com` redirects to `springboard.google.com` and then redirect to `cloudsearch.google.com`

fast recon GG (from global to granular) by `omespino@googleESCAL8:~/london$`

asset brute forcing

Basic process, use `all.txt` and fuzz any web application
in this case `https://springboard.google.com`

wfuzz - The Web Fuzzer @xmendez

Wfuzz has been created to facilitate the task in web applications assessments and it is based on a simple concept: it replaces any reference to the FUZZ keyword by the value of a given payload.

```
time wfuzz -c -w all.txt https://springboard.google.com/FUZZ
```

```
Total requests - 2,286,549
Requests/sec.   - 484.21
Realtime:       - 78m42s ~ 1 hour 18 minutes
```

ffuf - Fuzz Faster U Fool @joohoi

A fast web fuzzer written in Go.

heavily inspired by the great projects gobuster and wfuzz.

```
time fuff -c -w all.txt -u https://springboard.google.com/FUZZ
```

```
Total requests - 2,286,549
Requests/sec.   - 2019
Realtime:       - 18m52s !!!
```

wfuzz < fuff
SPEED UP 4.2x w00t!

after the fuzzing I found just 1 interesting directory `"/java"`

fast recon GG (from global to granular) by `omespino@googleESCAL8:~/london$`

web scraping - "/java" = Google's internal framework

Basic process, **download** the whole web application html and **grep** for the win

httrack - **Xavier Roche**

HTTrack is a free and open-source Web crawler and offline browser, developed by Xavier Roche and licensed under the GNU General Public License Version 3. HTTrack allows users to download World Wide Web sites from the Internet to a local computer.

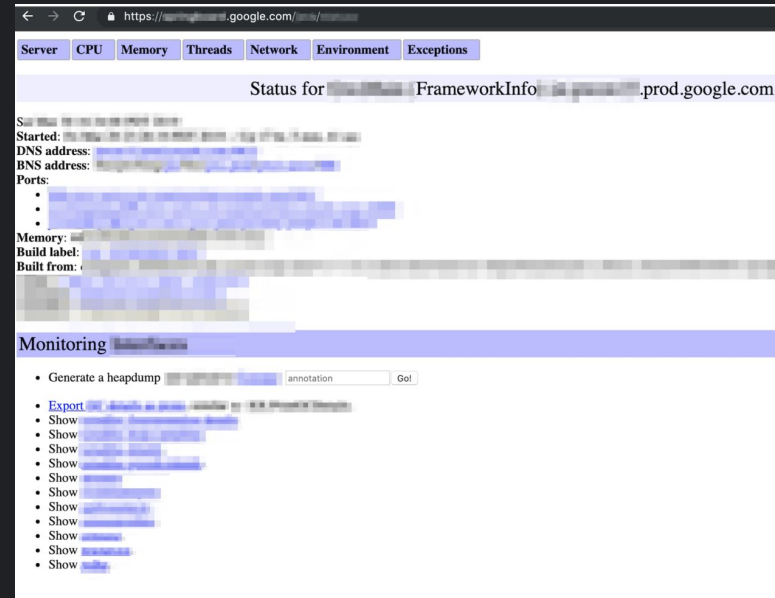
```
#first downloaded the whole framework page
httrack https://springboard.google.com/java/ -O
./springboard.google.com -v
```

```
#then grep the whole framework page (local mirror)
grep -rih "file=" springboard.google.com/ | grep env
```

```
[ - - - - - REDACTED - - - - - ]
https://springboard.google.com/https-cache/new.txt:18:05:27
4095/4095 ---MCZ 200 added ('OK') text/plain
date:Wed,%2016%20Oct%202019%2001:05:26%20GMT
```

https://springboard.google.com/java/procz?file=/proc/self/envron <---- w00000000t!!!!!!

```
[ - - - - - REDACTED - - - - - ]
```



fast recon GG (from global to granular) by omespino@googleESCAL8:~/london\$

web scraping - LFI on https://springboard.google.com

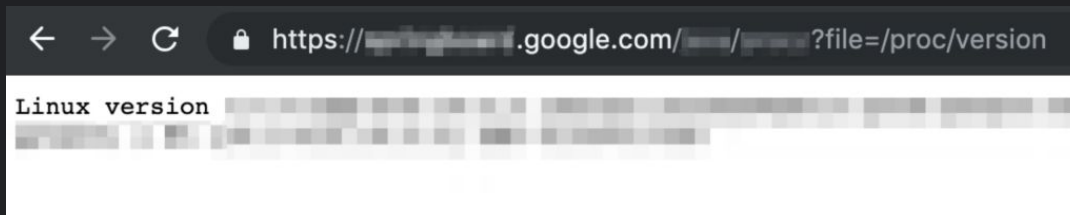
after get the url `https://springboard.google.com/java/procz?file=/proc/self/envron`
from grep, I open that url in the browser and ...



fast recon GG (from global to granular) by `omespino@googleESCAL8:~/london$`

web scraping - LFI on https://springboard.google.com

Just to be sure that was a full LFI working I tried to load another file, I checked with `/proc/version` and ...



To be honest I tried to escalate to RCE but I hadn't any success, since apparently it was very hardened, I wasn't able to read `/proc/*/fd`, ssh keys, server keys or any logs.

fast recon GG (from global to granular) by `omespino@googleESCAL8:~/london$`

LFI on <https://springboard.google.com> recap

subdomains filtering and screenshots
<https://springboard.corp.google.com>
was interesting because was a
forbidden site (works accounts only)

download the whole web app
"/java" with httrack and
grep "file=" | grep "env"

global organization
discovery:

- find the scope (rules)
- pick some domain

environment + top level domain
from the scope
corp.google.com



subdomain
discovery

visual
identification



assets brute
forcing

wfuzz with all.txt and
"/java" dir found
(google's internal
framework)



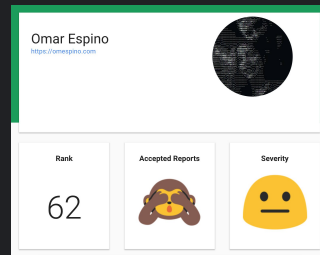
web scraping



PROFIT

RESULTS

- a lot of fun
- google's hall of fame 62th rank
- \$13,337 USD bounty



fast recon GG (from global to granular) by omespino@googleESCAL8:~/london\$

what's next? duck test - fingerprinting and repeat

"If it looks like a duck, swims like a duck, and quacks like a duck, then it probably is a duck."

shodan, oh my dear friend ❤️

one day I was thinking, what if there are more unprotected google's internal framework instances, then I just search:

`http.html:/redacted org:"Google"`

RESULTS

P	TYPE	TITLE	ASSIGNEE	STATUS
P3	Bug	other in google corp prod server	wo...@google.com	Accepted
P1	Bug	other in Google Server	wo...@google.com	Accepted
P1	Bug	other in Google Server	wo...@google.com	Accepted
P4	Customer Issue	other in Google Server	--	Duplicate

The screenshot shows the Shodan search interface. The search bar contains the query `http.html:/redacted org:"Google"`. The results are categorized into 'TOTAL RESULTS' (6), 'TOP COUNTRIES' (United States), 'TOP SERVICES' (Oracle, HTTP), and 'TOP ORGANIZATIONS' (Google Cloud). On the right, there are three detailed service entries for Google Cloud, each showing HTTP status (200 OK), content type (text/html), date, server information, and various headers like X-Goog-Netnon-Label and X-Goog-Security-Sign.

fast recon GG (from global to granular) by `omespino@googleESCAL8:~/london$`

lessons learned

```
omespino@googleESCAL8:~/london$ cat lessons_learned.txt
```

- keep it simple never underestimate the basics
- tweak your tools timeouts / threads
- be professional
- discipline - be constant
- do not rush - it take times
- avoid burnouts - go outside, hang up with family and friends
- learn how communicate - learn english
- read read read read read read (twitter #bugbounty #writeup)
- be patient - 16 mo wait reports o more
- repeat

Q&A

fast recon GG (from global to granular) by [omespino@google](#)ESCAL8:~/london\$

ありがとう
ARIGATŌ

GRAZIE

DANKE

DANKON

धन्यवाद

THANK YOU

MERCI

谢谢
XIÈXIÈ

감사합니다

СПАСИБО

GRACIAS