ATTACKING ORGANIZATIONS

FROM ZERO TO HERO

HUSSEIN DAHER

INTRODUCTION

- Hussein Daher
- CEO of **WebImmunify.com**
- over 10 years of cybersecurity experience
- business master's degree
- over 1000 vulnerabilities on bug bounty platforms
- H1-2010's Vigilante award
- Intigriti's 1337up competition #1
- Udemy course, "Bug Bounty - An Advanced Guide to Finding Good Bugs," - 1800 students

IMPORTANCE OF RECONNAISSANCE

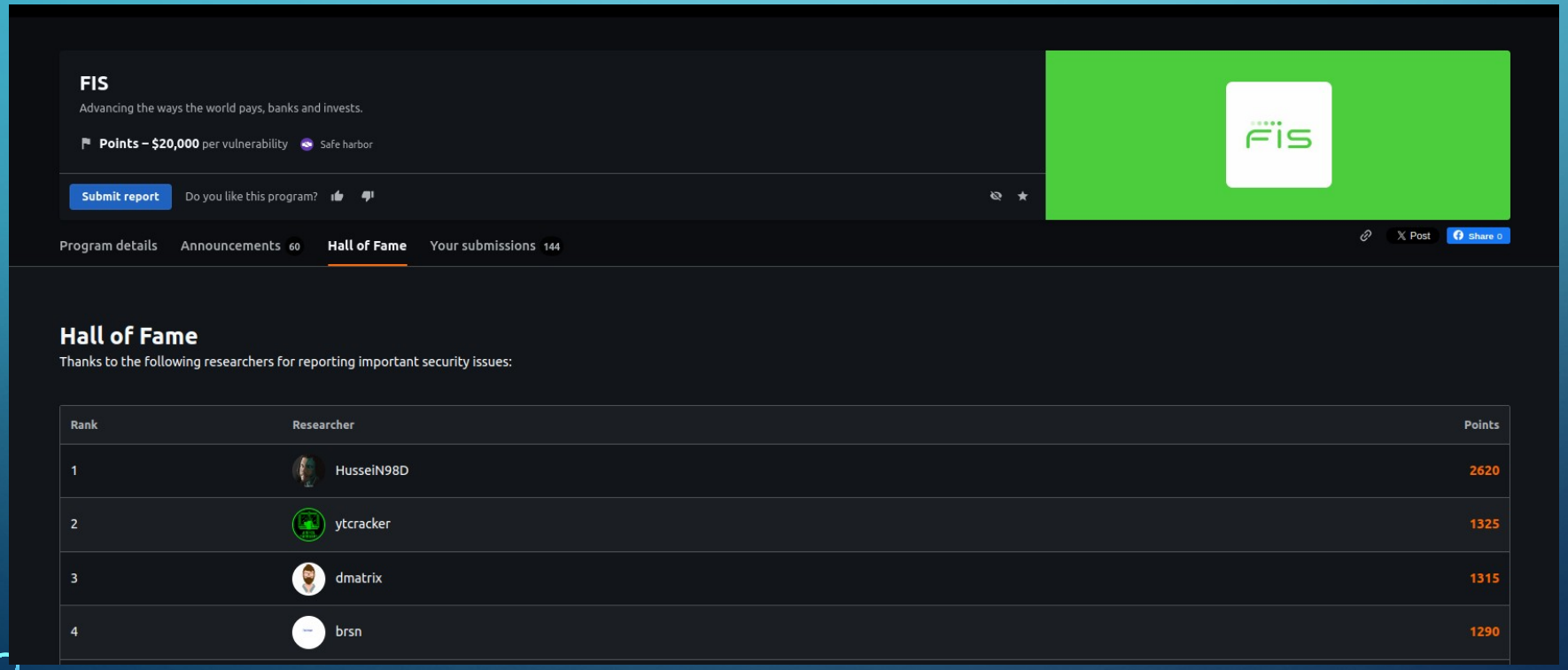
- Reconnaissance is the first and one of the most critical steps in the bug bounty hunting process. It involves gathering as much information as possible about the target to identify potential weaknesses and entry points. Effective reconnaissance can significantly enhance the chances of finding valuable bugs.

IDENTIFYING POTENTIAL ATTACK SURFACES

- Attack surfaces are the various points in a system where an unauthorized user can try to enter data to or extract data from. Identifying these surfaces involves looking at the entire digital footprint of an organization, including web applications, network infrastructure, APIs, and more.

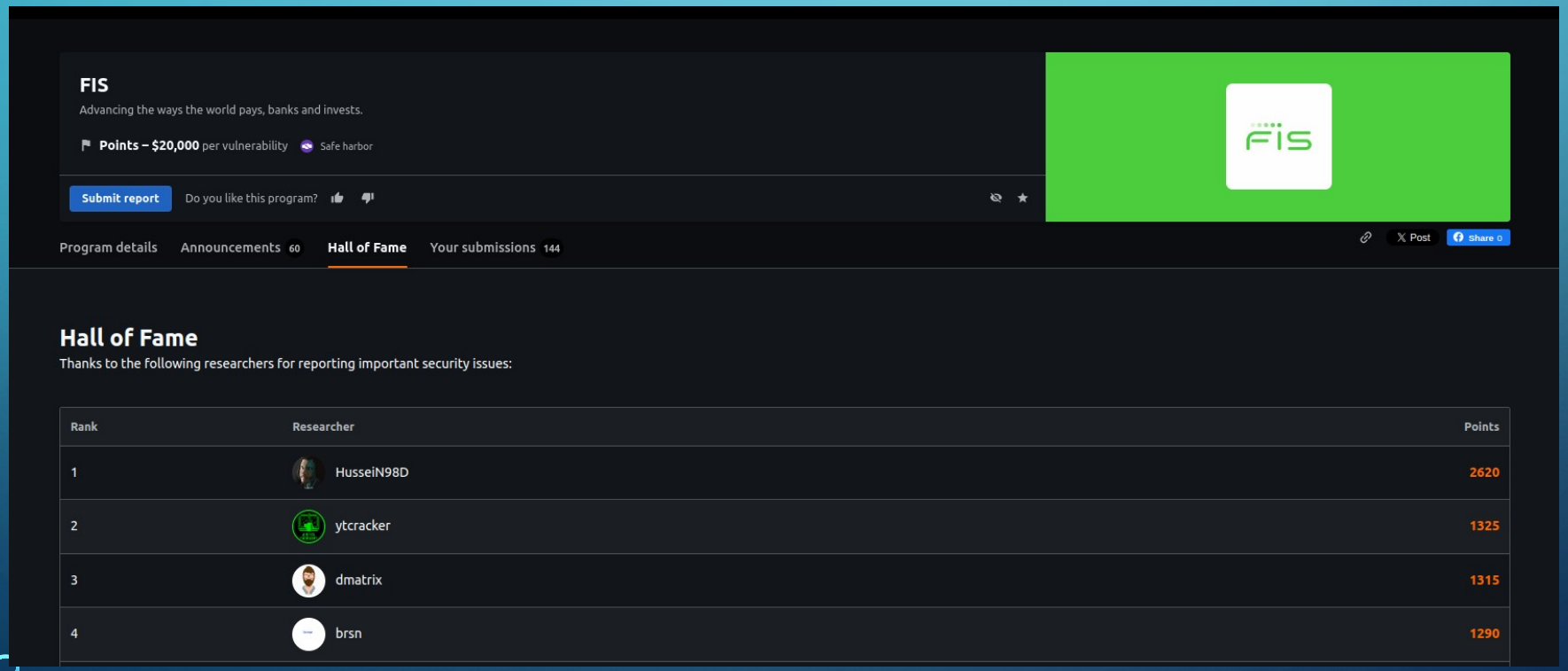
INTRO

- I’ve been able to take over FIS #1 spot in under 6 months with perseverance in recon combined with application hacking skills



INTRO

- FIS is a large organization and their bug bounty program is about “everything they own” in scope. This gives us a huge attack surface to look at.



METHODOLOGIES

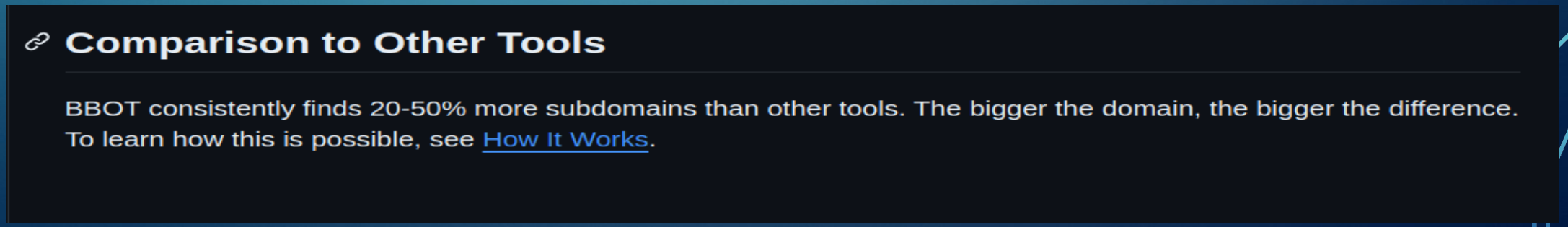
- Subdomain Enumeration
- VHOST Identification
- ASN Mapping
- Web Fuzzing
- Dorking
- Other tips and tricks

SUBDOMAIN ENUMERATION

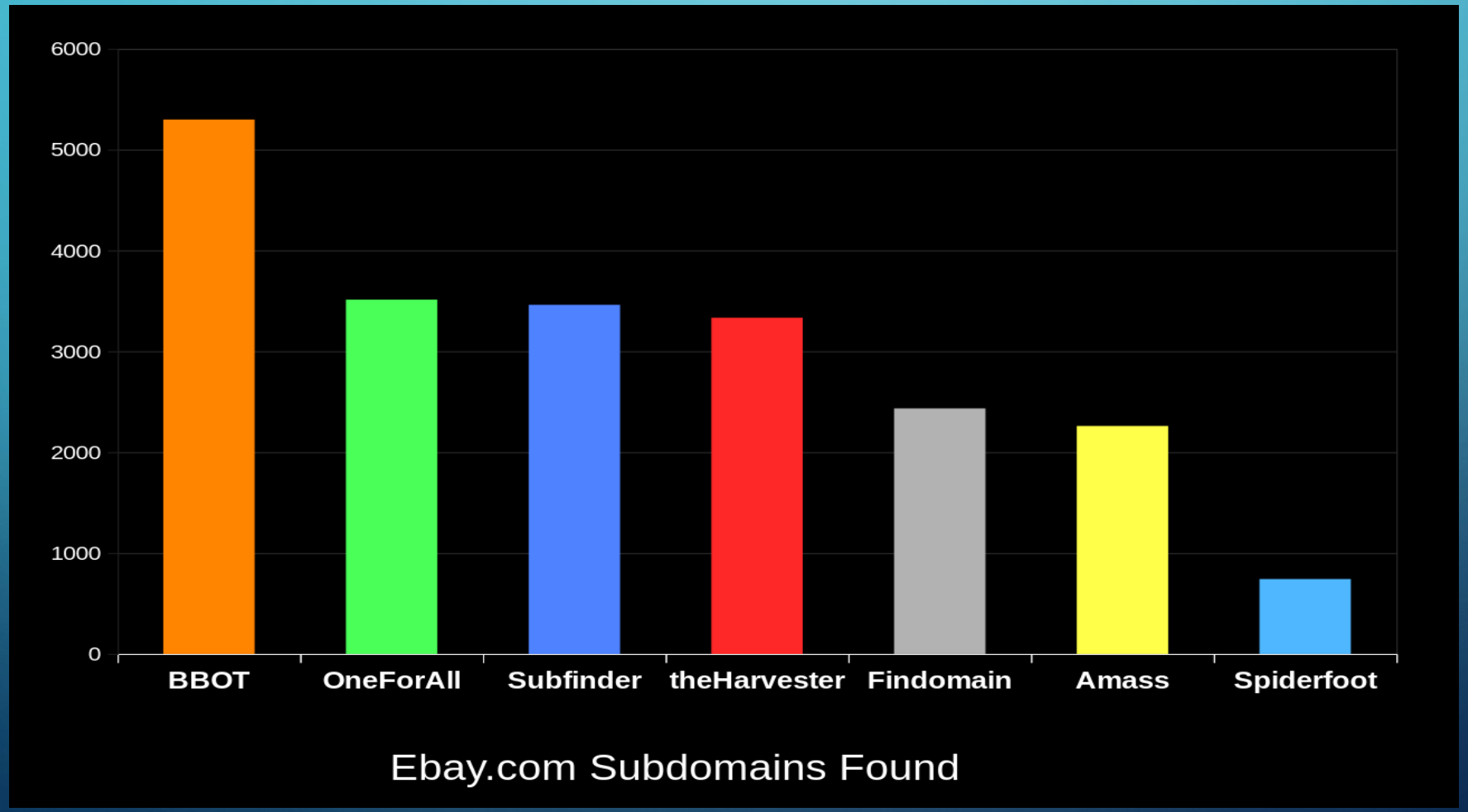
- Identifying subdomains will give you a bigger attack surface
- Look for preprod/env subdomains
- Perform Recursive bruteforcing:
- FUZZ.host → dev.host → FUZZ.dev.host

SUBDOMAIN ENUMERATION

- The good old scripts:
- amass enum -passive -d example.com -o results.txt
- sublist3r -d example.com
- Add to your toolset:
- https://github.com/blacklanternsecurity/bbot



BBOT COMPARISON

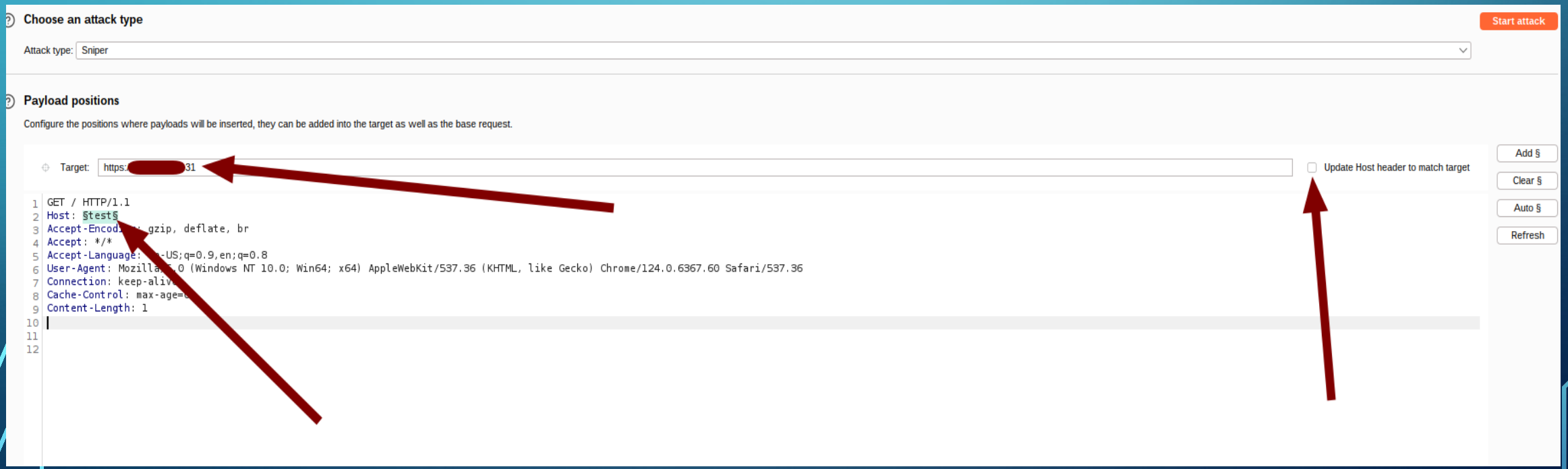


VHOST IDENTIFICATION

- Less people deep into VHOSTS
- Identifying these will give you targets most people haven’t seen yet
- Spot the differences

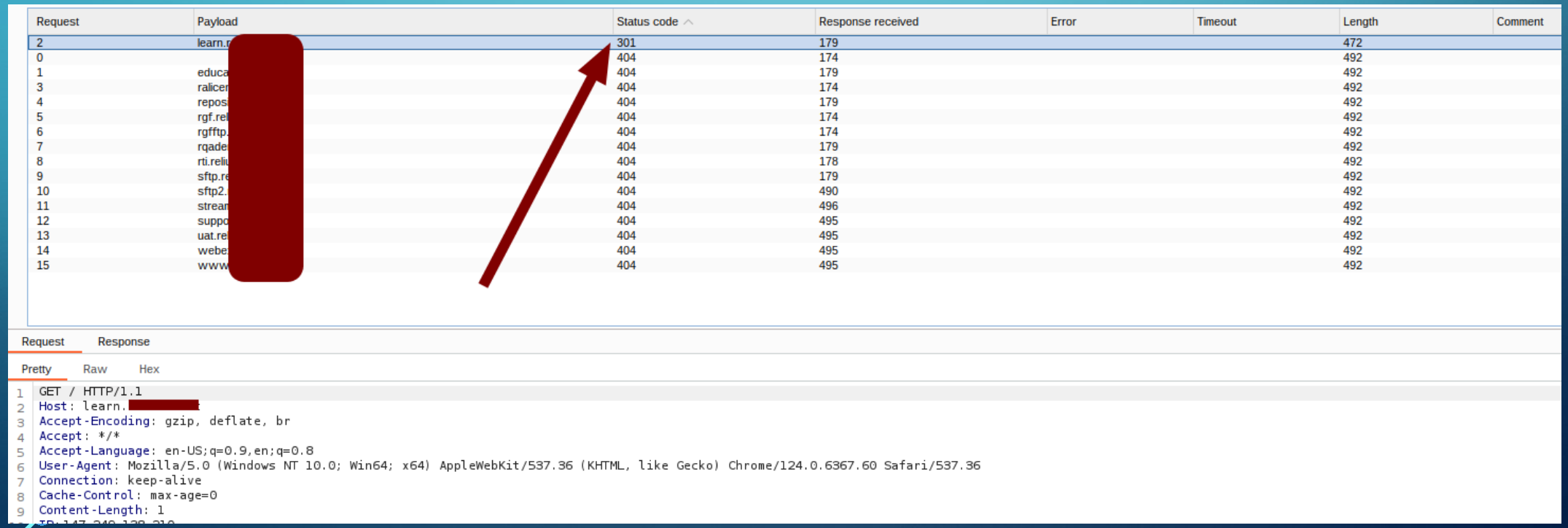
VHOST IDENTIFICATION

- What I love doing : BurpSuite Intruder



VHOST IDENTIFICATION

- Results



Request	Payload	Status code	Response received	Error	Timeout	Length	Comment
2	learn.[redacted]	301	179			472	
0		404	174			492	
1	educa[redacted]	404	179			492	
3	ralicer[redacted]	404	174			492	
4	reposi[redacted]	404	179			492	
5	rgf.rel[redacted]	404	174			492	
6	rgfftp.[redacted]	404	174			492	
7	rqade[redacted]	404	179			492	
8	rti.relia[redacted]	404	178			492	
9	sftp.re[redacted]	404	179			492	
10	sftp2.[redacted]	404	490			492	
11	strean[redacted]	404	496			492	
12	suppo[redacted]	404	495			492	
13	uat.re[redacted]	404	495			492	
14	webe[redacted]	404	495			492	
15	www[redacted]	404	495			492	

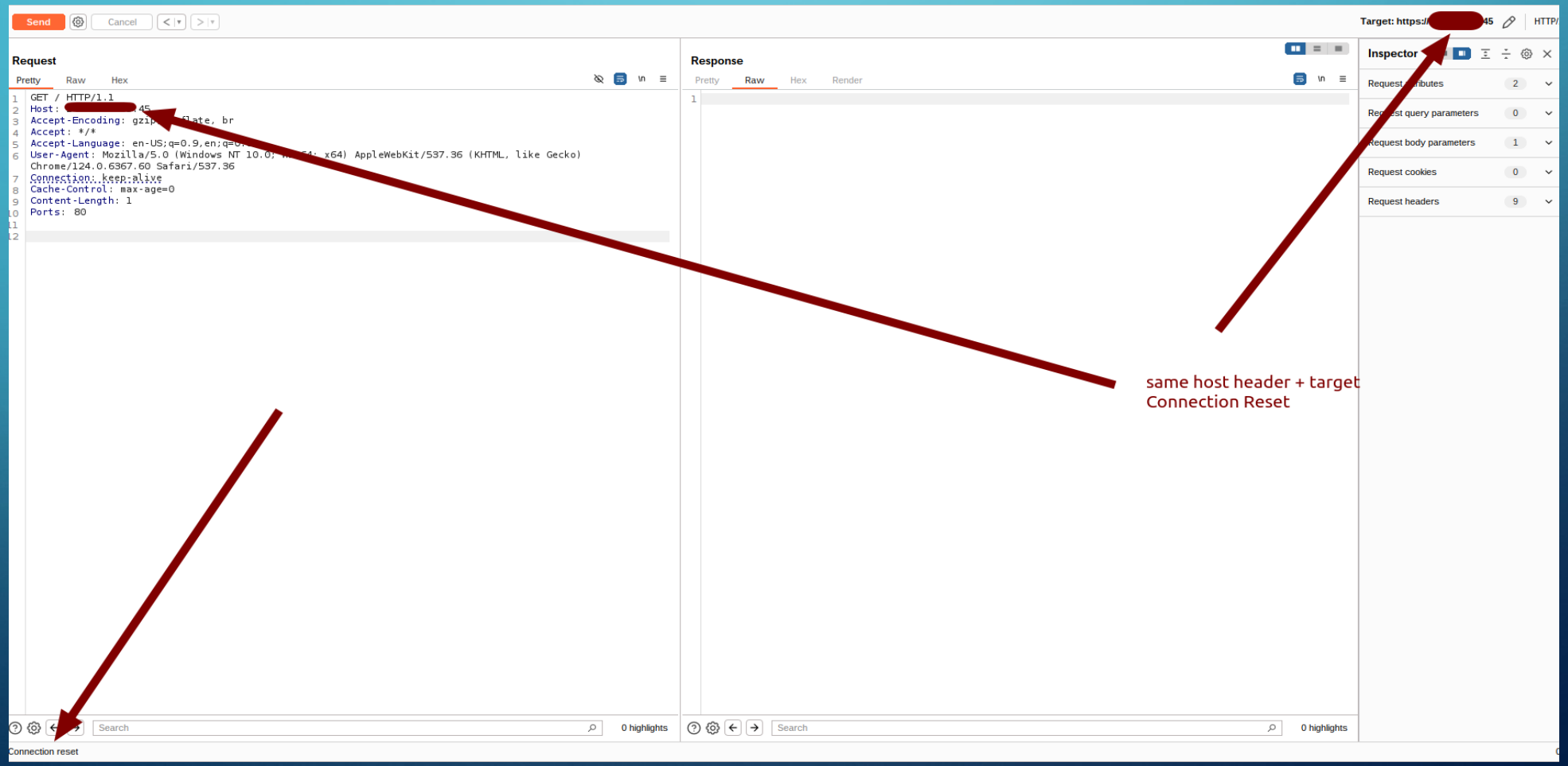
Request Response

Pretty Raw Hex

```
1 GET / HTTP/1.1
2 Host: learn.[redacted]
3 Accept-Encoding: gzip, deflate, br
4 Accept: */*
5 Accept-Language: en-US;q=0.9,en;q=0.8
6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0.6367.60 Safari/537.36
7 Connection: keep-alive
8 Cache-Control: max-age=0
9 Content-Length: 1
```

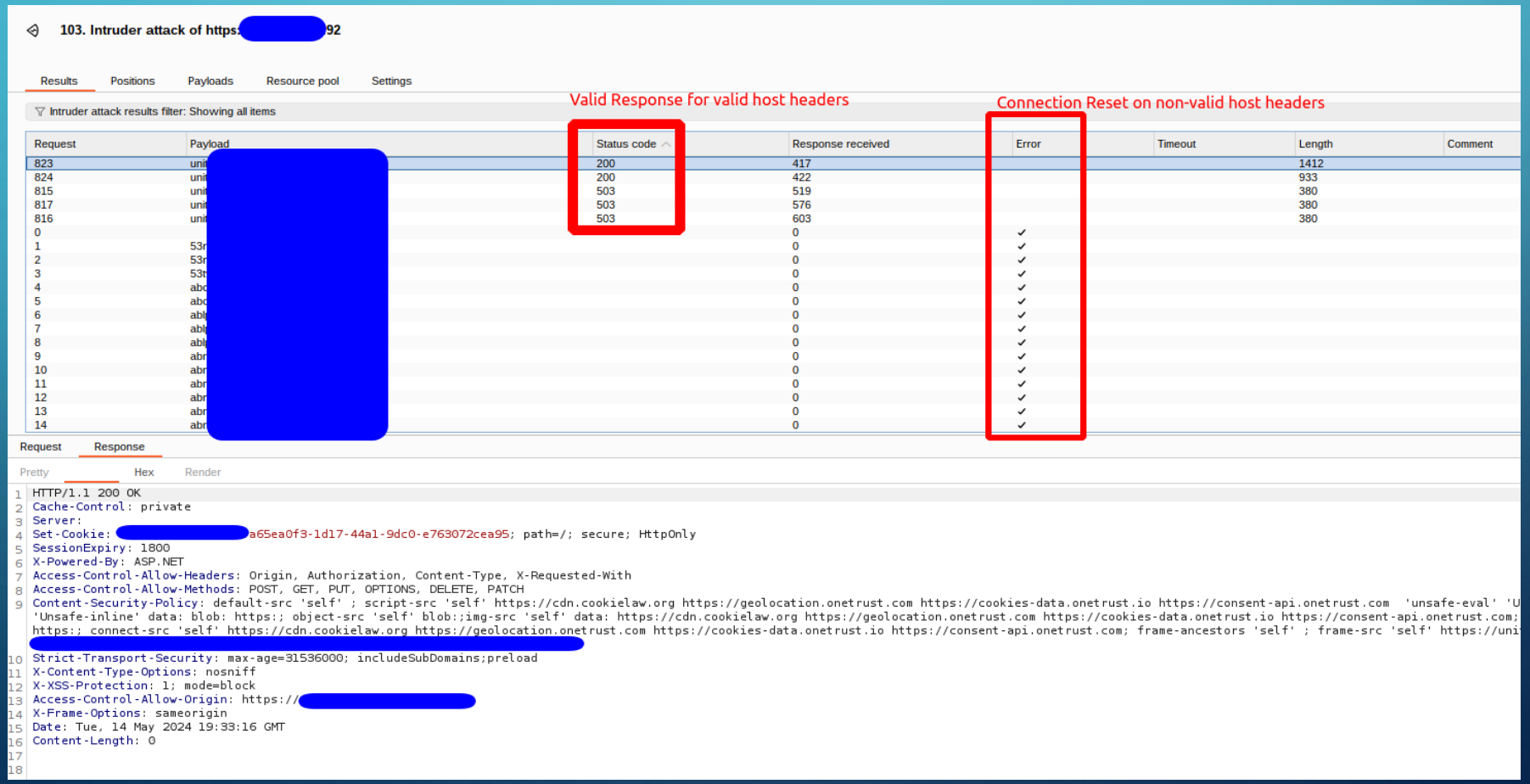
VHOST IDENTIFICATION

- More



VHOST IDENTIFICATION

- However, bruteforcing host header gives us more targets



ASN MAPPING

- https://bgp.he.net/search?search%5Bsearch%5D=Facebook.+Inc&commit=Search

AS63293	ASN	Facebook, Inc.
AS54115	ASN	Facebook Inc
AS34825	ASN	FACEBOOK ISRAEL LTD
AS32934	ASN	Facebook, Inc.
AS149642	ASN	Facebook Singapore Pte Ltd.
74.119.76.0/22	Route	Facebook, Inc.
69.63.184.0/21	Route	Facebook, Inc.
69.63.176.0/21	Route	Facebook, Inc.
69.63.176.0/20	Route	Facebook, Inc.
69.171.250.0/24	Route	Facebook, Inc.
69.171.240.0/20	Route	Facebook, Inc.
69.171.224.0/20	Route	Facebook, Inc.
69.171.224.0/19	Route	Facebook, Inc.
66.220.152.0/21	Route	Facebook, Inc.
66.220.144.0/21	Route	Facebook, Inc.
66.220.144.0/20	Route	Facebook, Inc.
45.64.40.0/22	Route	Facebook Singapore Pte Ltd.
41.223.111.0/24	Route	for the above IPv4 we are using for google and facebook peering
41.189.185.0/24	Route	MTN Ghana Enterprise Internet Clients and facebook Cache
2c0f:ef78:f::/48	Route	Facebook South Africa (Pty) Ltd
2c0f:ef78:e::/48	Route	Facebook South Africa (Pty) Ltd
2c0f:ef78:d::/48	Route	Facebook South Africa (Pty) Ltd
2c0f:ef78:12::/48	Route	Facebook South Africa (Pty) Ltd
2c0f:ef78:11::/48	Route	Facebook South Africa (Pty) Ltd
2c0f:ef78:10::/48	Route	Facebook South Africa (Pty) Ltd
2620:10d:c09b::/48	Route	Facebook Inc
2620:10d:c09a::/48	Route	Facebook Inc
2620:10d:c099::/48	Route	Facebook Inc
2620:10d:c098::/48	Route	Facebook Inc

ASN MAPPING

- Copy all IP ranges

```
hussein98d@HusseiN98D:/tmp$ cat iplist | cut -f1
129.134.156.0/24
129.134.155.0/24
129.134.154.0/24
129.134.150.0/24
129.134.149.0/24
129.134.148.0/24
129.134.147.0/24
129.134.144.0/24
129.134.143.0/24
129.134.140.0/24
129.134.139.0/24
129.134.138.0/24
129.134.136.0/24
129.134.135.0/24
129.134.134.0/24
129.134.133.0/24
129.134.132.0/24
129.134.131.0/24
129.134.130.0/24
```

ASN MAPPING

- Send to prips

```
hussein98d@HusseiN98D:/tmp$ for i in $(cat iplist | cut -f1); do prips $i >> fb; done
hussein98d@HusseiN98D:/tmp$ wc -l fb
5120 fb
hussein98d@HusseiN98D:/tmp$ head -n 10 fb
129.134.156.0
129.134.156.1
129.134.156.2
129.134.156.3
129.134.156.4
129.134.156.5
129.134.156.6
129.134.156.7
129.134.156.8
129.134.156.9
hussein98d@HusseiN98D:/tmp$
```

ASN MAPPING

- Gather all subdomains you can find for the org, even the ones not resolving

```
hussein98d@HusseiN98D:/tmp$ cat fbsubs | awk -F' ' '{print $2}' | sort -u
edgeray-fna-msgr-shv-01-fktw4.fbcdn.net
edgeray-fna-shv-01-fktw4.fbcdn.net
fna-fbcdn-shv-01-fktw4.fbcdn.net
fna-fbcdn-video-shv-01-fktw4.fbcdn.net
fna-fbcdn-z-m-shv-01-fktw4.fbcdn.net
fna-fbcdn-z-p3-shv-01-fktw4.fbcdn.net
fna-fbcdn-z-p4-shv-01-fktw4.fbcdn.net
fna-instagram-p42-shv-01-fktw4.fbcdn.net
fna-instagram-p4-shv-01-fktw4.fbcdn.net
fna-instagram-shv-01-fktw4.fbcdn.net
fna-internal-services-shv-01-fktw4.fbsv.net
fna-whatsapp-shv-01-fktw4.fbcdn.net
```

ASN MAPPING

- BruteForce : IPS:SUBDOMAINS

```
hussein98d@HusseiN98D:/tmp$ for i in $(cat fbips);do ffuf -w fbsubs -u https://$i -H "Host: FUZZ" -of csv -o $i.csv ; done

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v2.1.0-dev
________________________________________________

 :: Method           : GET
 :: URL              : https://129.134.156.0
 :: Wordlist         : FUZZ: /tmp/fbsubs
 :: Header           : Host: FUZZ
 :: Output file      : 129.134.156.0.csv
 :: File format      : csv
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200-299,301,302,307,401,403,405,500
________________________________________________

:: Progress: [15/15] :: Job [1/1] :: 0 req/sec :: Duration: [0:00:00] :: Errors: 0 ::
```

WEB FUZZING

- Brute force using FFUF
- CRAWL using Katana
- Use archives such as waybackmachine
- Use URL shortners

WEB FUZZING

- Create custom wordlist for the application
- 1- gather all urls you can find of the target using katana, gau and others. Save all into one file and sort uniq

```
hussein98d@HusseiN98D:/tmp$ gau example.com
https://mautic.example.com/
http://mautic.example.com/
http://www.example.com/5
http://policy-explanation.example.com
http://azwbqj95.example.com/
https://example.com/image/inspirational_quote.jpg
https://example.com/audio/inspirational_quote.mp3
http://example.com/..
http://www.example.com/source/caminho/para/script.php
https://geoserver.example.com/coredb
http://geoserver.example.com/
https://billing.example.com/discord.php&client_id=&client_secret=
https://billing.example.com/discord.php&client_id=1162861284455362570&client_secret
http://billing.example.com/
http://example.com/Main Page
http://www.example.com/some_img.jpg
https://scope.example.com/
http://scope.example.com/
http://scope2.example.com/
https://example.com/method/mymethod
http://example.com/method/mymethod
http://example.com/method/foo
https://example.com/method/foo
https://example.com/recordings/stream.mp4
https://link.example.com
```

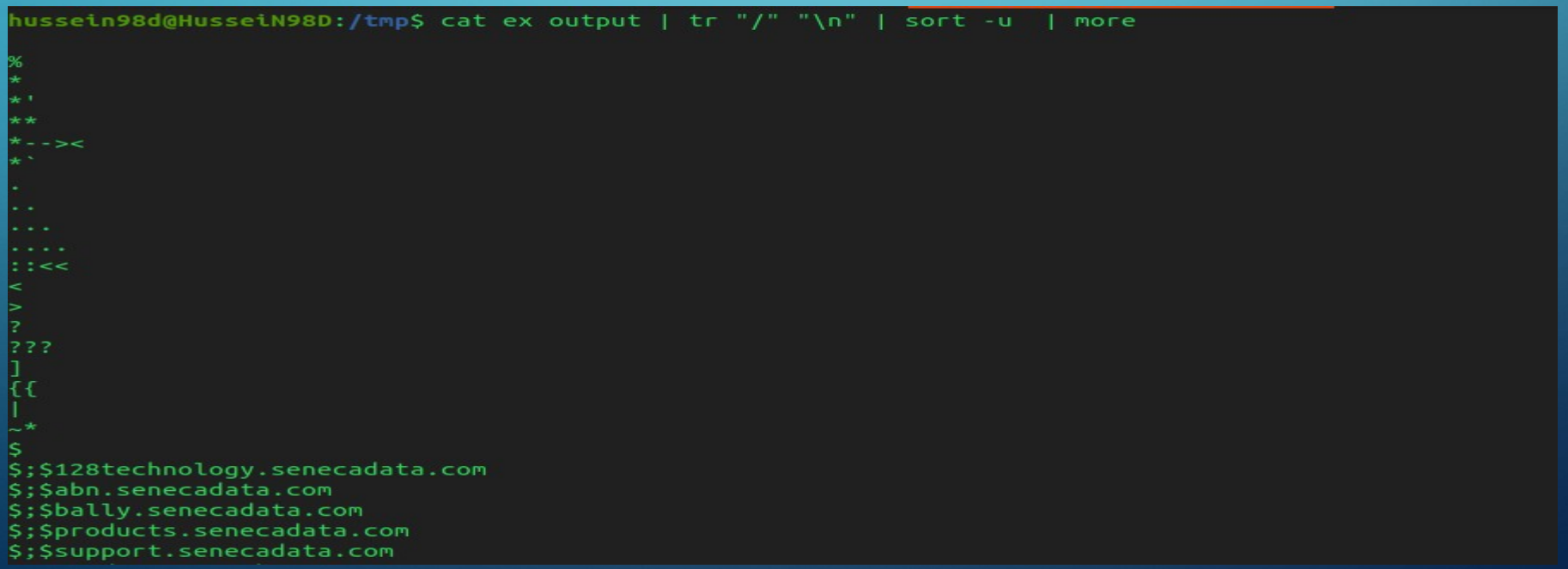
WEB FUZZING

- Now, run LinkFinder on all of the URLs https://github.com/GerbenJavado/LinkFinder

```
hussein98d@HusseiN98D:/tmp$ cat ex | rush -j10 "python3 /home/hussein98d/Desktop/Tools/LinkFinder/linkfinder.py -o cli
-i {} | sort -u >> output"
```

WEB FUZZING

- Finally , sort uniq URLs + Endpoints found from LinkFinder and Crawling and create the wordlist



WEB FUZZING

- FUZZ all subdomains/Ips of the org with this wordlist
- Keep adding new paths/file names found

DORKING

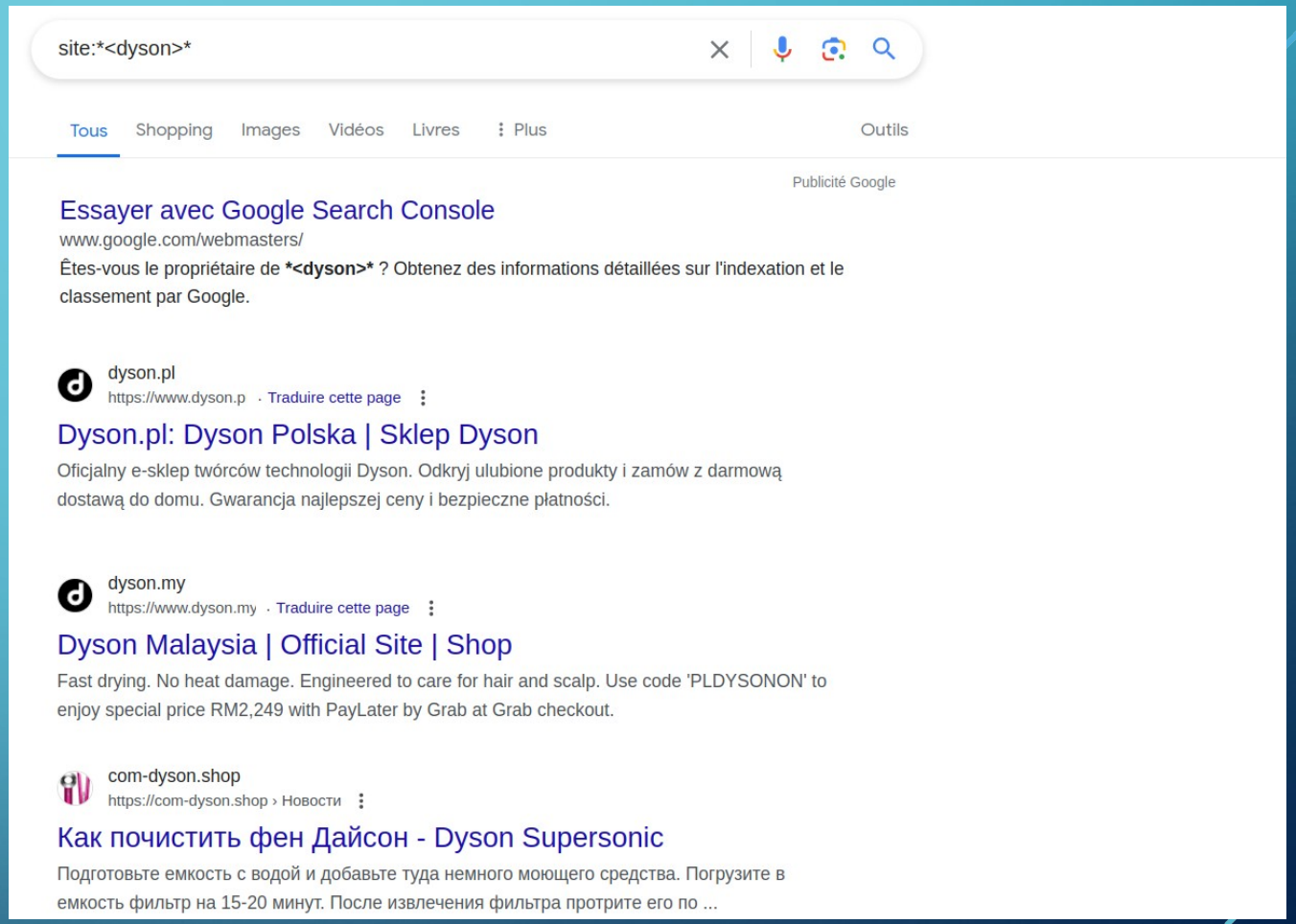
- Be creative when using dorks
- Use multiple search engines (Google, Duck, Bing, etc)
- Create your own dorks

DORKING

- Powerful less known dorks
- site:*<example>*

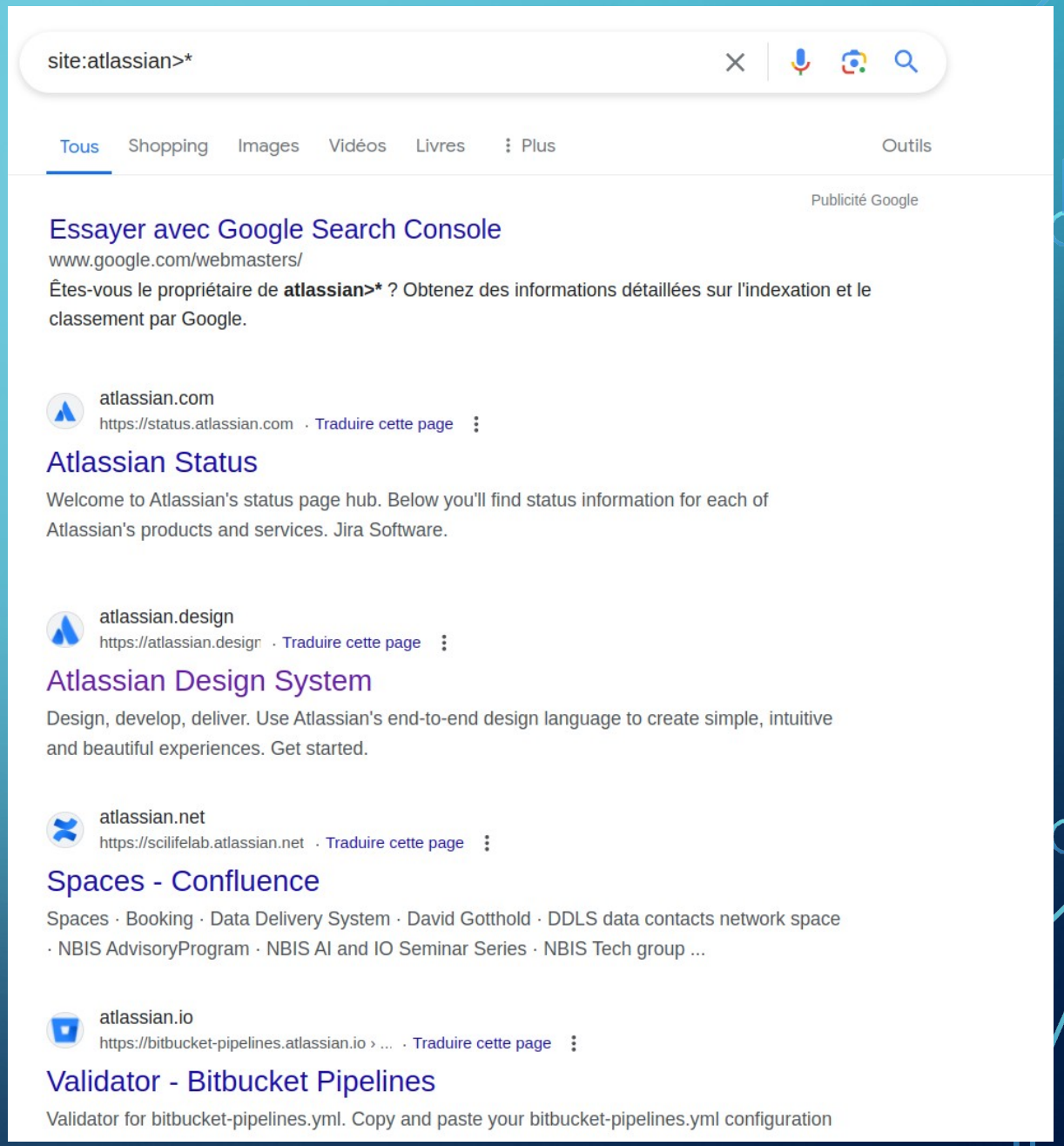
site:: This operator restricts the search results to a specific site or domain.

<example>: The asterisks (*) are wildcards that match any character(s). In this case, the dork will match any domain or subdomain that contains the word "example".



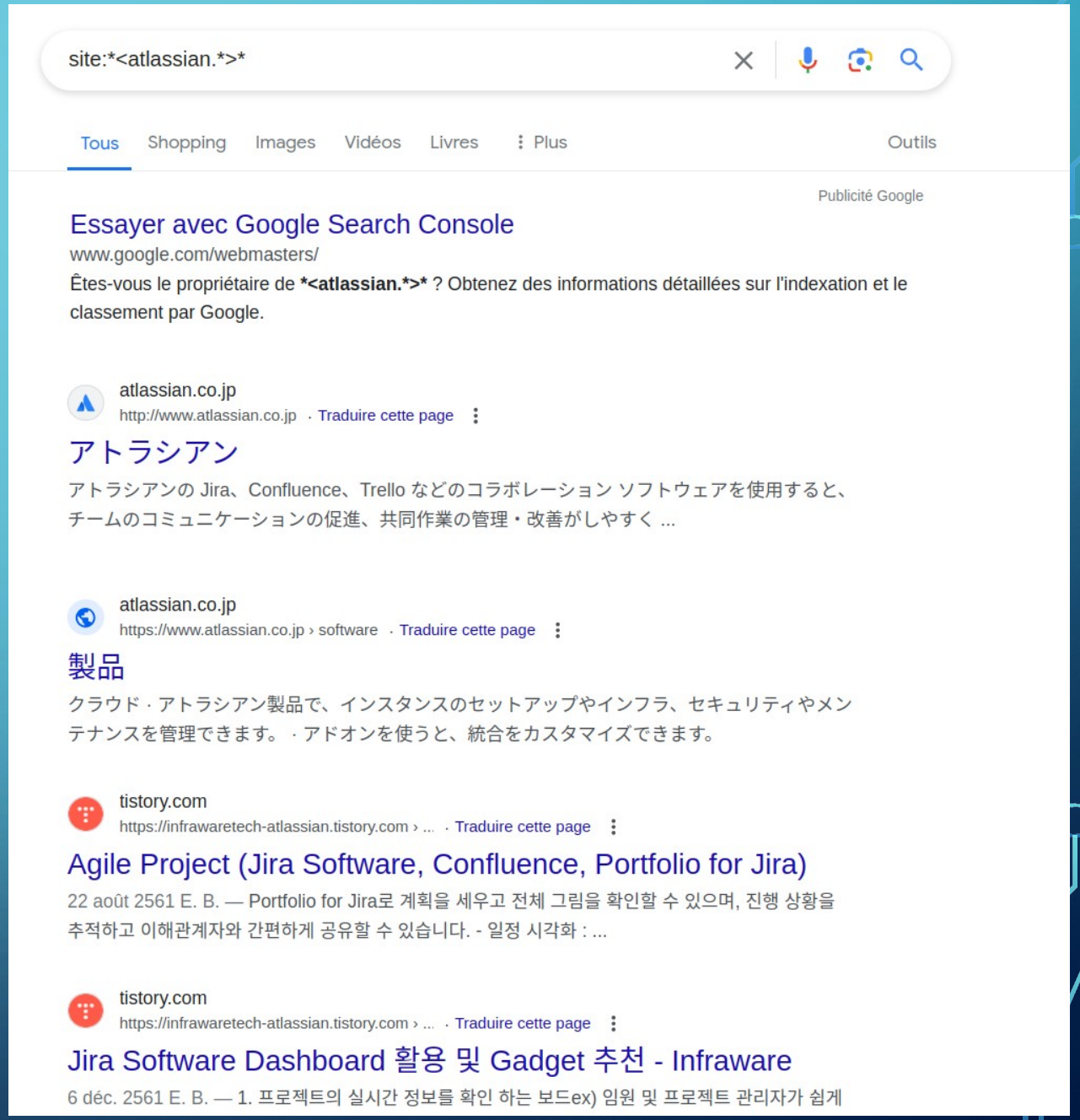
DORKING

- Powerful less known dorks
- site:atlassian>*



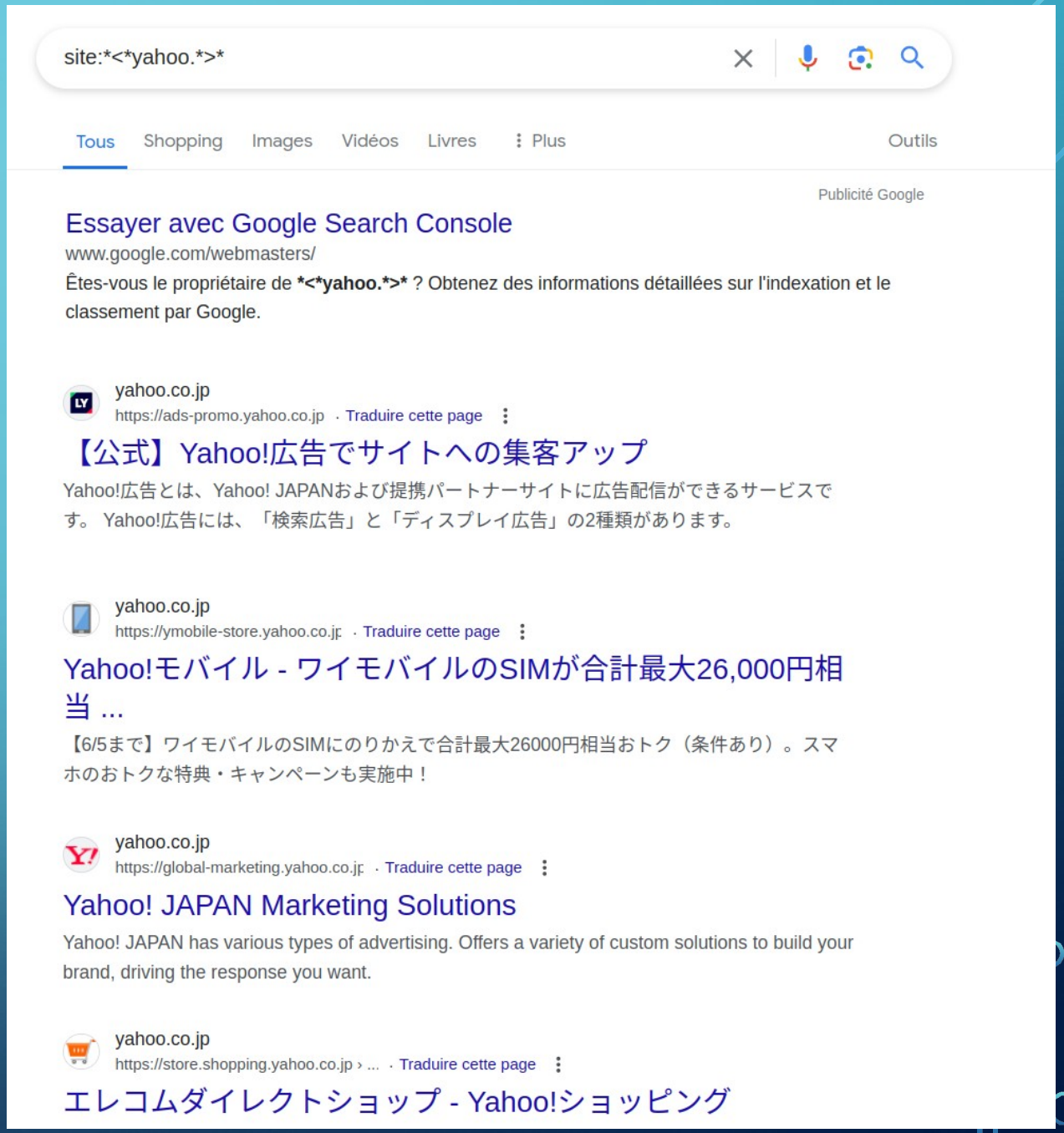
DORKING

- Powerful less known dorks
- site:*<atlassian.*>*



DORKING

- Powerful less known dorks
- site:*<*yahoo.*>*

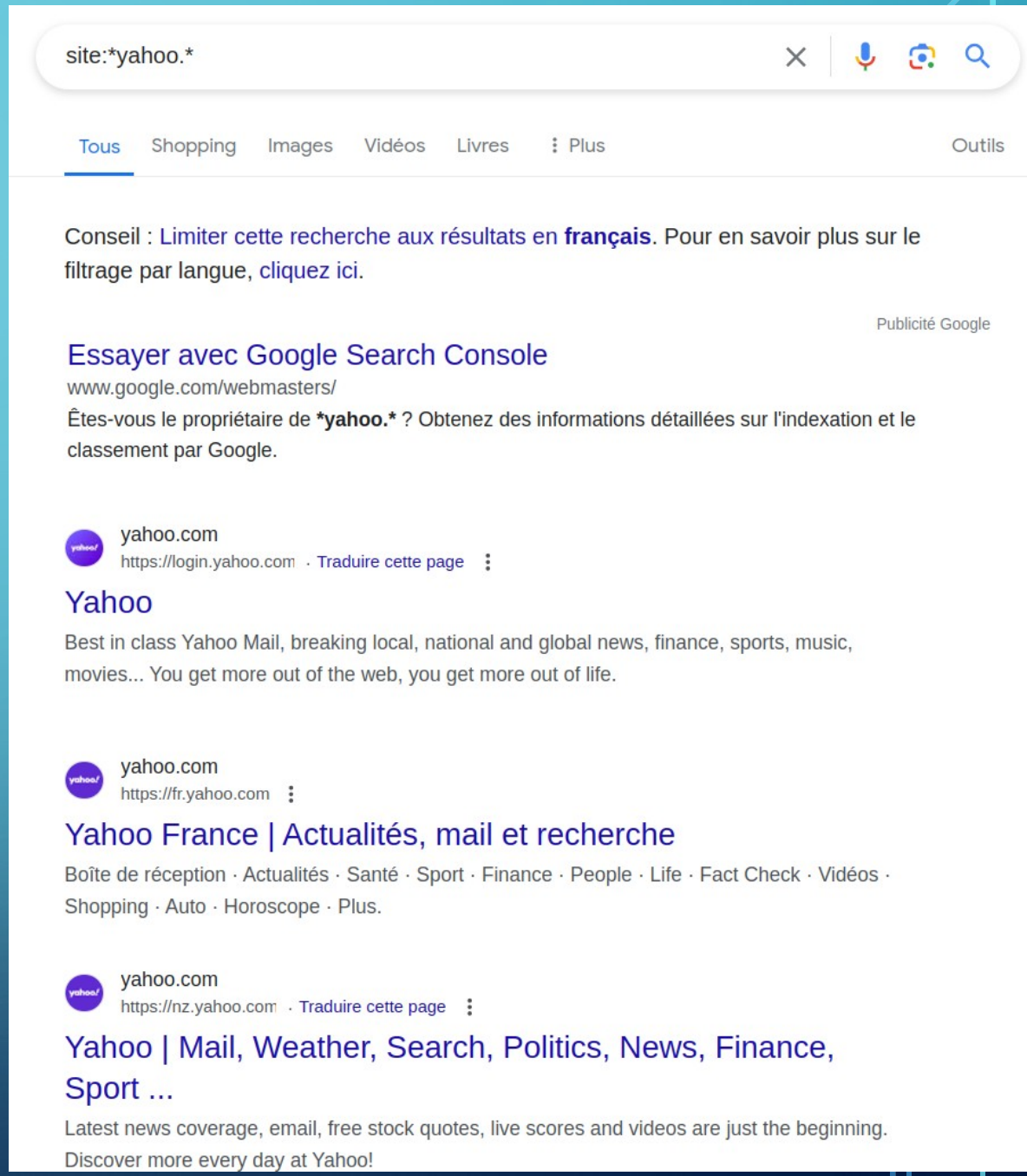


DORKING

- Powerful less known dorks
- site:*yahoo.*

site:: This operator restricts the search results to a specific site or domain.

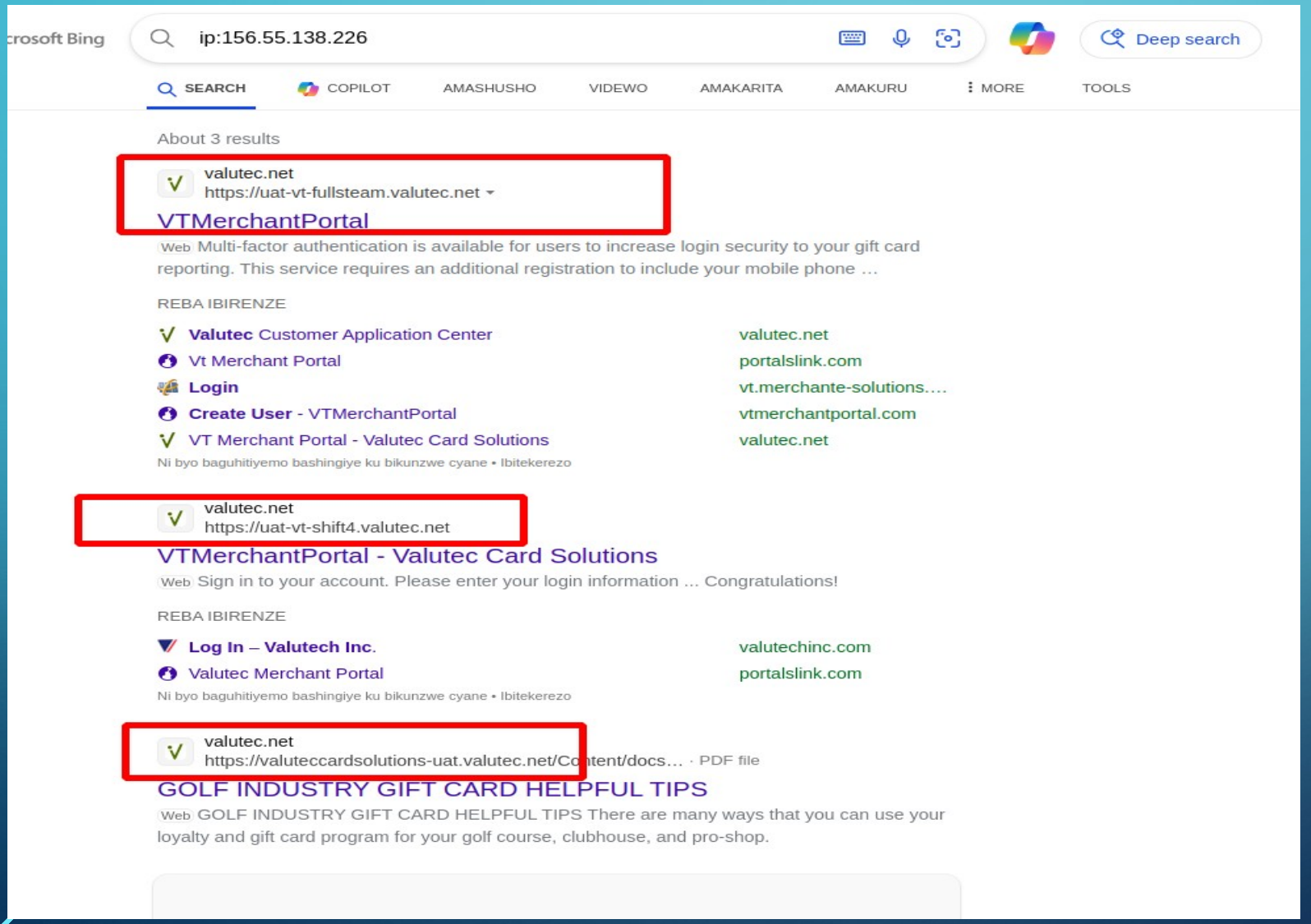
yahoo.: The asterisks (*) are wildcards that match any character(s). In this case, the dork will match any subdomain and any top-level domain associated with Yahoo.



BING

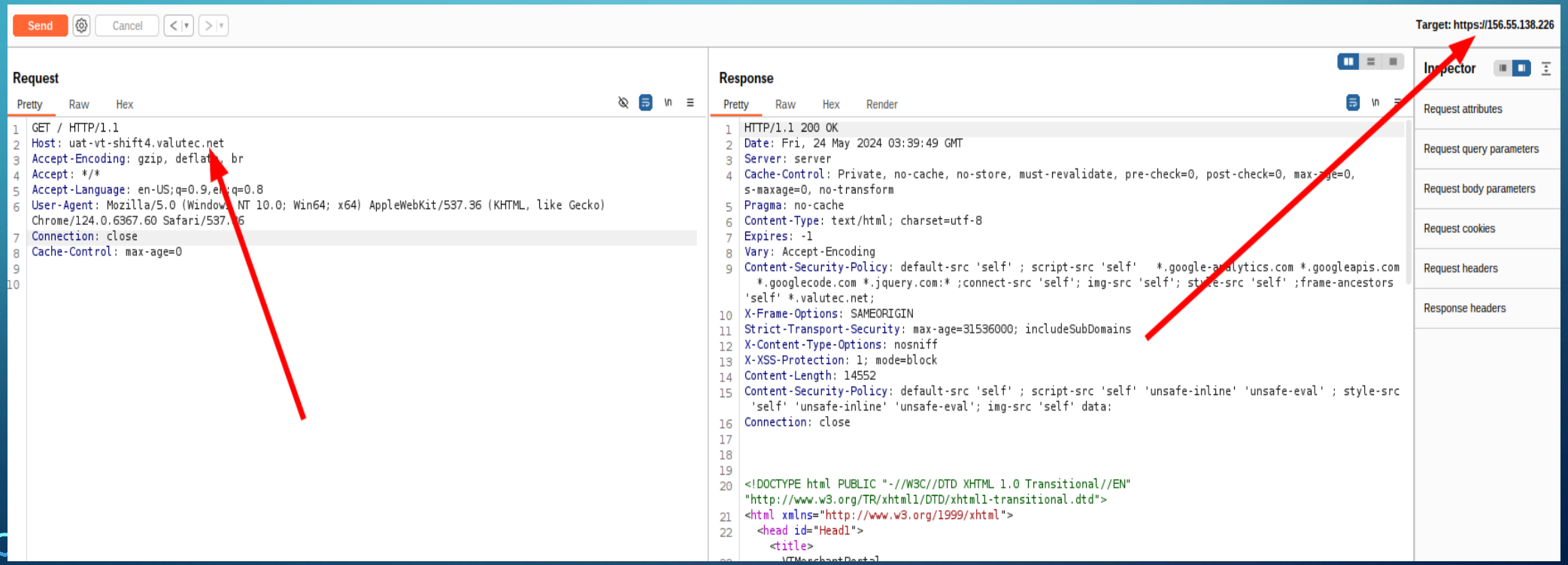
- Remember the IP list we got from ASN?
- Use bing to find valid hosts on the server
- Dork: “ip:127.0.0.1”

BING


crosoft Bing
ip:156.55.138.226
Deep search
SEARCH
COPILOT
AMASHUSHO
VIDEWO
AMAKARITA
AMAKURU
MORE
TOOLS
About 3 results
valutec.net
https://uat-vt-fullsteam.valutec.net
VTMerchantPortal
Web Multi-factor authentication is available for users to increase login security to your gift card reporting. This service requires an additional registration to include your mobile phone ...
REBA IBIRENZE
Valutec Customer Application Center
valutec.net
Vt Merchant Portal
portalslink.com
Login
vt.merchante-solutions....
Create User - VTMerchantPortal
vtmerchantportal.com
VT Merchant Portal - Valutec Card Solutions
valutec.net
Ni byo baguhitiyemo bashingiye ku bikunzwe cyane • Ibitekerezo
valutec.net
https://uat-vt-shift4.valutec.net
VTMerchantPortal - Valutec Card Solutions
Web Sign in to your account. Please enter your login information ... Congratulations!
REBA IBIRENZE
Log In – Valutech Inc.
valutechinc.com
Valutec Merchant Portal
portalslink.com
Ni byo baguhitiyemo bashingiye ku bikunzwe cyane • Ibitekerezo
valutec.net
https://valuteccardsolutions-uat.valutec.net/Content/docs... · PDF file
GOLF INDUSTRY GIFT CARD HELPFUL TIPS
Web GOLF INDUSTRY GIFT CARD HELPFUL TIPS There are many ways that you can use your loyalty and gift card program for your golf course, clubhouse, and pro-shop.

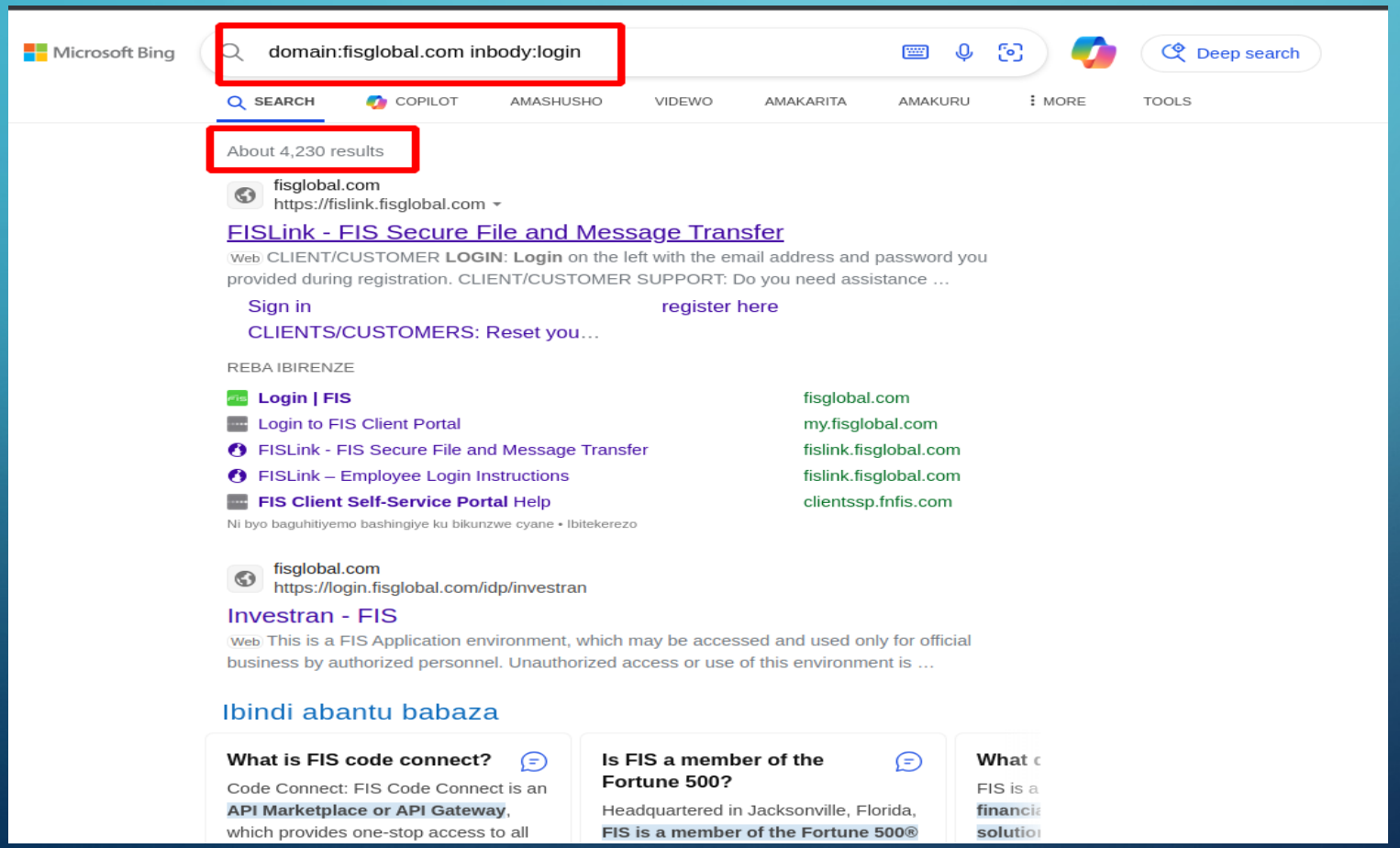
BING

- Sometimes the DNS name will not resolve . In this case, send IP + Domain name returned by Bing



BING

- Use inbody:example instreamset:(title url):example

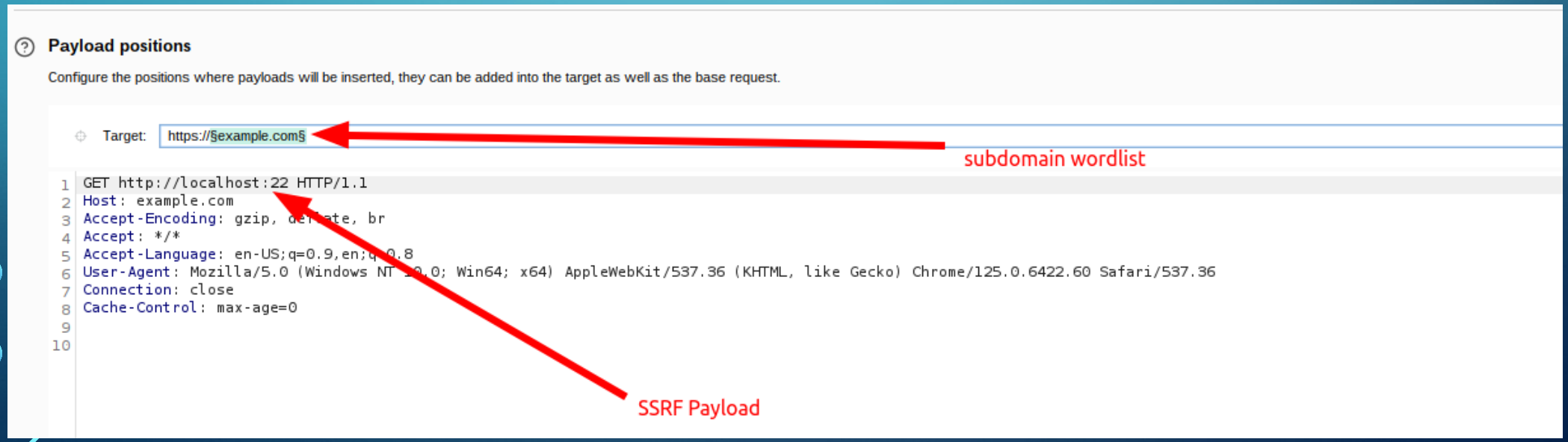


TIPS AND TRICKS

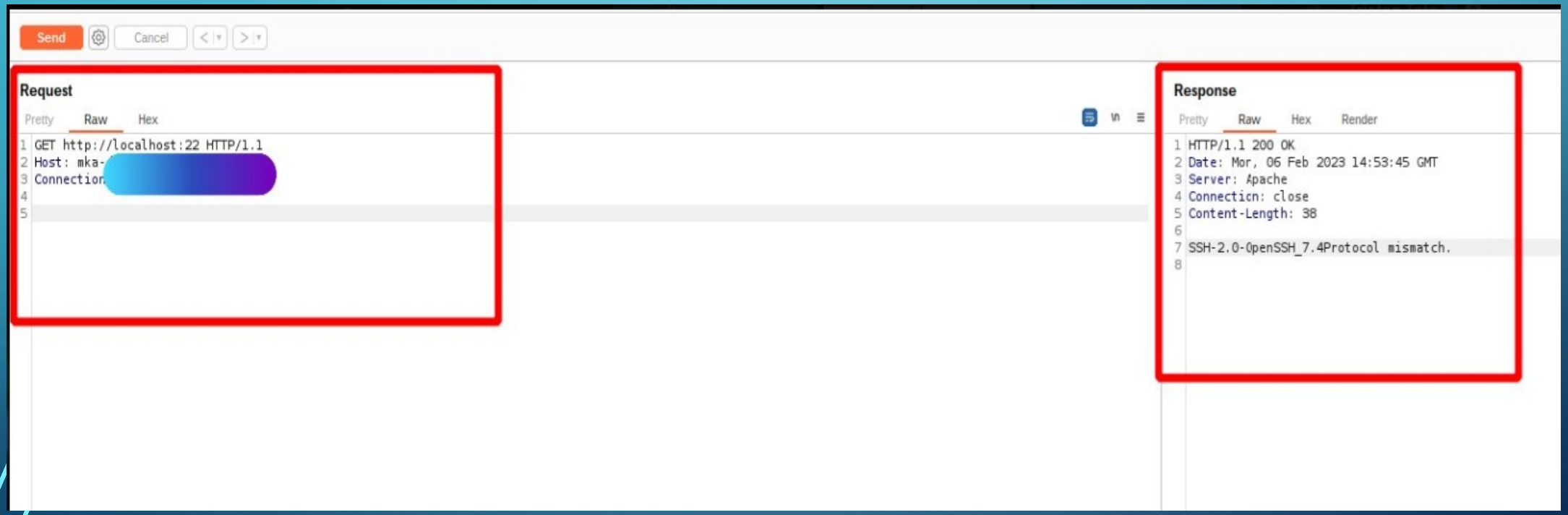
- Gather all URLs of target using all techniques discussed
- Sent to Intruder and mass-test
- Example: GET http://localhost:22 HTTP/2

TIPS AND TRICKS

- Gather all URLs of target using all techniques discussed
- Sent to Intruder and mass-test
- Example: GET http://localhost:22 HTTP/2

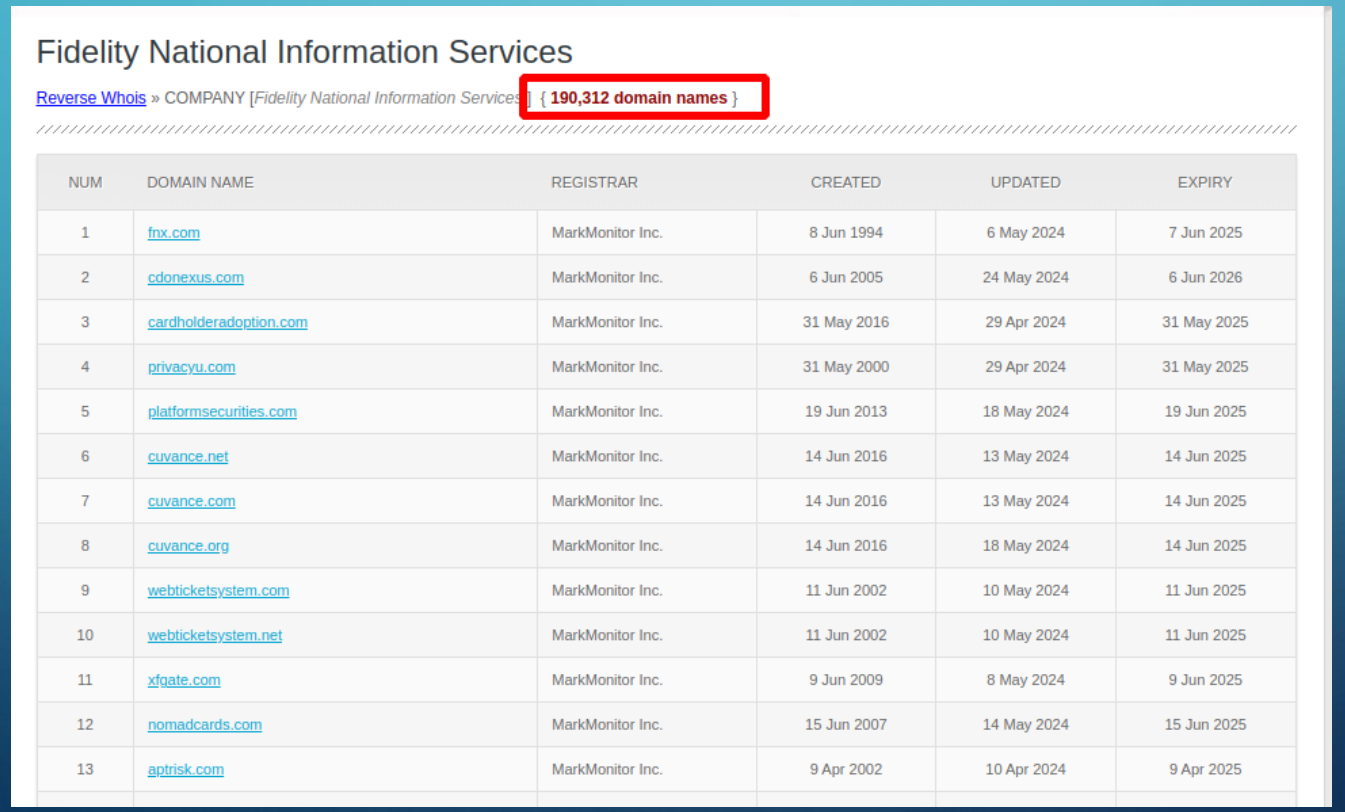


TIPS AND TRICKS



CASE STUDY

- When I started to hack on FIS program, I found tons of domain names they own by using reverse whois: https://www.whoxy.com/company/41024



Fidelity National Information Services

Reverse Whois » COMPANY [*Fidelity National Information Services*] { **190,312 domain names** }

NUM	DOMAIN NAME	REGISTRAR	CREATED	UPDATED	EXPIRY
1	fnx.com	MarkMonitor Inc.	8 Jun 1994	6 May 2024	7 Jun 2025
2	cdonexus.com	MarkMonitor Inc.	6 Jun 2005	24 May 2024	6 Jun 2026
3	cardholderadoption.com	MarkMonitor Inc.	31 May 2016	29 Apr 2024	31 May 2025
4	privacyu.com	MarkMonitor Inc.	31 May 2000	29 Apr 2024	31 May 2025
5	platformsecurities.com	MarkMonitor Inc.	19 Jun 2013	18 May 2024	19 Jun 2025
6	cuvance.net	MarkMonitor Inc.	14 Jun 2016	13 May 2024	14 Jun 2025
7	cuvance.com	MarkMonitor Inc.	14 Jun 2016	13 May 2024	14 Jun 2025
8	cuvance.org	MarkMonitor Inc.	14 Jun 2016	18 May 2024	14 Jun 2025
9	webticketsystem.com	MarkMonitor Inc.	11 Jun 2002	10 May 2024	11 Jun 2025
10	webticketsystem.net	MarkMonitor Inc.	11 Jun 2002	10 May 2024	11 Jun 2025
11	xfgate.com	MarkMonitor Inc.	9 Jun 2009	8 May 2024	9 Jun 2025
12	nomadcards.com	MarkMonitor Inc.	15 Jun 2007	14 May 2024	15 Jun 2025
13	aptrisk.com	MarkMonitor Inc.	9 Apr 2002	10 Apr 2024	9 Apr 2025

CASE STUDY

- It would be impossible for me to dork all these domains manually
- Decided to take the hard way and code a tool to bypass search engines captchas, rate limits , and others

CASE STUDY

- It would be impossible for me to dork all these domains manually
- Decided to take the hard way and code a tool to bypass search engines captchas, rate limits , and others

CASE STUDY

- After spending a couple days coding my tool, I finally came up with a tool to dork on Google, Bing, Yahoo, DuckDuckgo and Baidu in no time

```
root@localhost:/tmp# bash dork.sh
Usage: dork.sh <dork> [--engines engine1 engine2 ...]

Arguments:
  <dork>                    The dork query to search for.
  --engines engine1 engine2  (Optional) Specify the search engines to use (default: aol baidu google duck bing).
                            Available engines: aol, baidu, google, duck, bing.
Example:
  dork.sh "inurl:admin" --engines google bing
root@localhost:/tmp# rm dork.sh ; nano dork.sh
root@localhost:/tmp# bash dork.sh
Usage: dork.sh <dork> [--engines engine1 engine2 ...]

Arguments:
  <dork>                    The dork query to search for.
  --engines engine1 engine2  (Optional) Specify the search engines to use (default: aol baidu google duck bing).
                            Available engines: aol, baidu, google, duck, bing.
Example:
  dork.sh "inurl:admin" --engines google bing
coded by @hussein98d
root@localhost:/tmp#
```

CASE STUDY

- My tool would take as input the dork and the engines to search on

```
root@localhost:/tmp# bash dork.sh "site:fisglobal.com 'login'"  --engines google bing > fisout
root@localhost:/tmp# cat fisout | httplinks | wc -l
391
root@localhost:/tmp# cat fisout | head -n 20
295
95
1445
595
1095
1495
1845
895
1795
40
80
1195
40
30
545
30
745
120
190
60
root@localhost:/tmp# cat fisout | httplinks| head -n 20
http://empower2.fisglobal.com/rs/092-EMI-875/images/MultiPay
https://acbs-docs-public-pnc.fisglobal.com/
https://acttst.fisglobal.com/
https://acttst.fisglobal.com/idpInitGlobalLogout.html
https://advisorgroup-sms.fisglobal.com/BackEnd/WebObjects/BackEnd.woa/wa/
https://advisorgroup-sms.fisglobal.com/Client/WebObjects/Client.woa/wa/
https://agla-sms.fisglobal.com/
https://aisecure-lr.fisglobal.com/
https://amer-pts-dr.fisglobal.com/
https://app.fcm.fisglobal.com/
https://aps2-sftp.imcs.fisglobal.com/EFTClient/Account/LostUsername.htm
https://aps-sftp.imcs.fisglobal.com/
https://ap-uat3.fisglobal.com/
https://bankofchinadrcitrix.fisglobal.com/
https://brinksvision1.fisglobal.com/VArchiveWebClient/Login.aspx
https://btps-hedgesuite-uat.fisglobal.com/
https://careers.fisglobal.com/us/en
https://careers.fisglobal.com/us/en/c/call-center-jobs?from=10
https://careers.fisglobal.com/us/en/c/finance-accounting-jobs
https://careers.fisglobal.com/us/en/c/sales-jobs?from=10
root@localhost:/tmp#
```

CASE STUDY

- Now, we have a dork tool working and a huge list of domains. What next? Dork with all sites!

```
root@localhost:/tmp/fis# head -n 10 list
infinity.com
olaccess.com
perfplusk12.com
sungarddx.com
automatedfinancial.com
securitiesinterlink7.com
automatedfinancial.com
benefitwebaccess.com
benefitwebaccess.net
infinity.com
root@localhost:/tmp/fis# for i in $(cat list); do bash dork.sh "site:$i 'login'" >> all
```

CASE STUDY

- After a couple permutations on dork and on all sites - example: site:example ‘login’ , site:example ‘upload’ etc I was left with around 100,000 unique links

```
root@localhost:/tmp/fis# cat out* | sort -u | httplinks | wc -l
99266
root@localhost:/tmp/fis#
```

CASE STUDY

- And just like this, we have a ton to look into

```
root@localhost:/tmp/fis# cat outall | grep -i login | head -n 30
http://demo-coe.automatedfinancial.com/login.html
http://ebtedge.com/cardholderlogin/
http://ebtedge.com/ezLogin.htm
http://ebtedge.com/login
http://ezcardinfo.com/login
http://infinity.com/login
http://login.billsupport.com/
http://retirementlogin.com/
http://retirementlogin.net/
http://retirementlogin.net/efc/help/ENG/contents.htm
http://retirementlogin.net/retirementllc/help/ENG/contents.htm
https://abacusbank.coreshare.insidefsi.com/Login.aspx
https://achievacu.coreshare.insidefsi.com/Login.aspx
https://achievacu.coreshare.insidefsi.com/Login_Auth.aspx
https://adamscommunity.coreshare.insidefsi.com/login.aspx
https://admin.regulatoryu.com/cgi/login.cgi
https://akron-oh.perfplusk12.com/login_process.asp
https://alpinesecurities.fisglobal.com/investor/login/oauth2/authorization/idp-wst
https://alpinesecurities.fisglobal.com/investor/ws/loginredirect
https://americanmetrobank.coreshare.insidefsi.com/Login.aspx
https://ameritas.infinity.com/login.aspx
https://ameritas.infinity.com/loginerror.aspx
https://ameritas.omniasp.com/loginerror.aspx
https://ameritas-temp.omniasp.com/login.aspx
https://anywhere-connections.fisglobal.com/login
https://anywhere-connections.fisglobal.com/login/authenticate
https://aon.infinity.com/login.aspx
https://app.biznowcard.com/login
https://app.fcm.fisglobal.com/account/login
https://app.nextgencontrol.com/Core/Authentication/Login
```

CASE STUDY

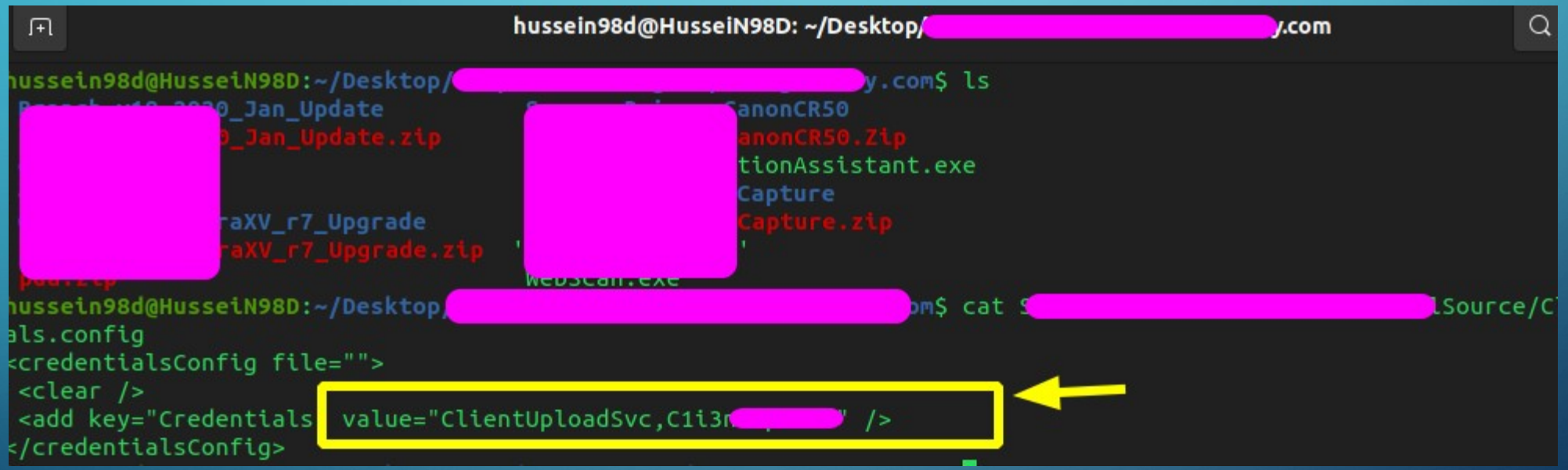
- All the discovered URLs can be again crawled using Katana to discover more and more endpoints/paths, and then again sent to LinkFinder, and then again back to katana, and then again LinkFinder .
- It’s a rinse and repeat process.
- Do not forget directory fuzzing on all sub-paths. For example hxxps://alpinesecurities.fisglobal.com/investor/login/oauth2/authorization/FUZZ

CASE STUDY

- During Mass-Dorking on my target, found a link similar to this:
- https://host/Documentation/Branch/archive.zip
- Upon downloading the archive, I found some credentials in a file

CASE STUDY

- Credentials:

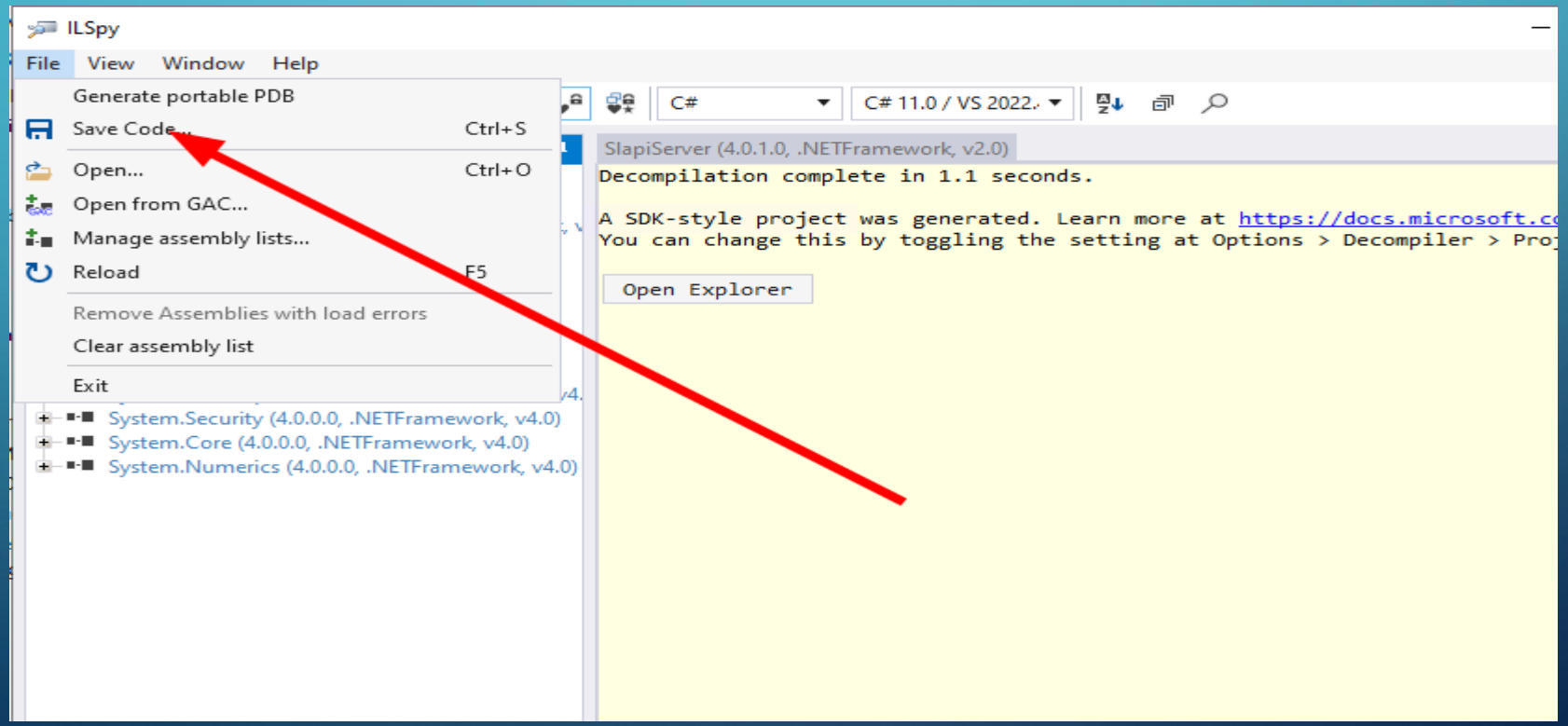


CASE STUDY

- Great, we have credentials – what next? What endpoint to hit with these credentials?
- Archive contained a .exe installer
- After installing the software on my VPS, I was left with a couple DLLs files

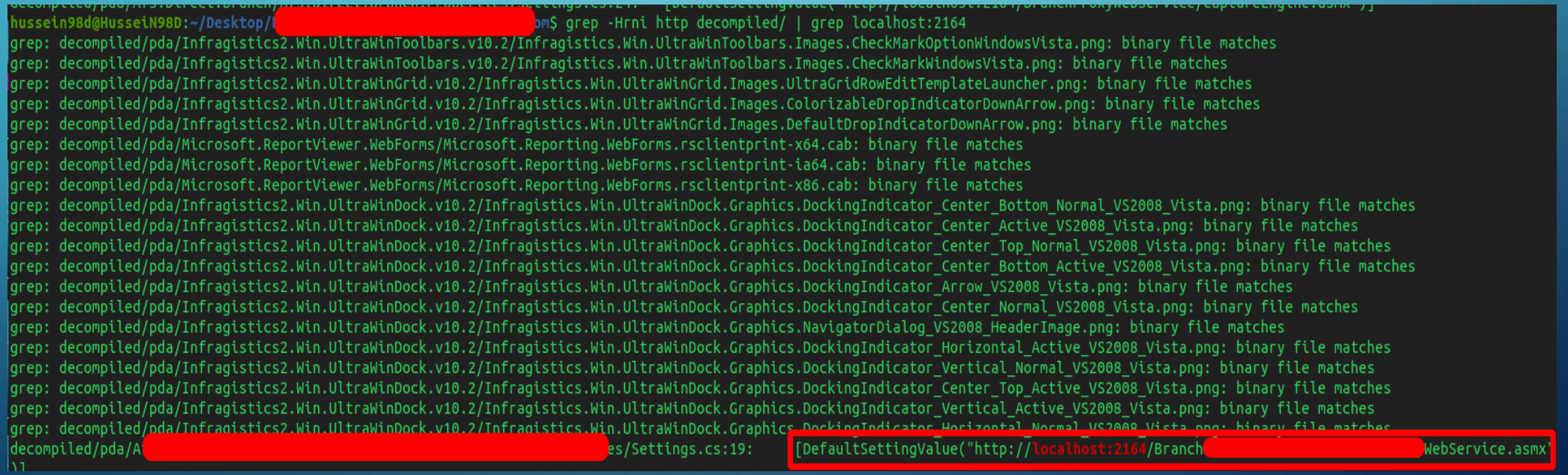
CASE STUDY

- Decompile using https://github.com/icsharpcode/ILSpy or Similar



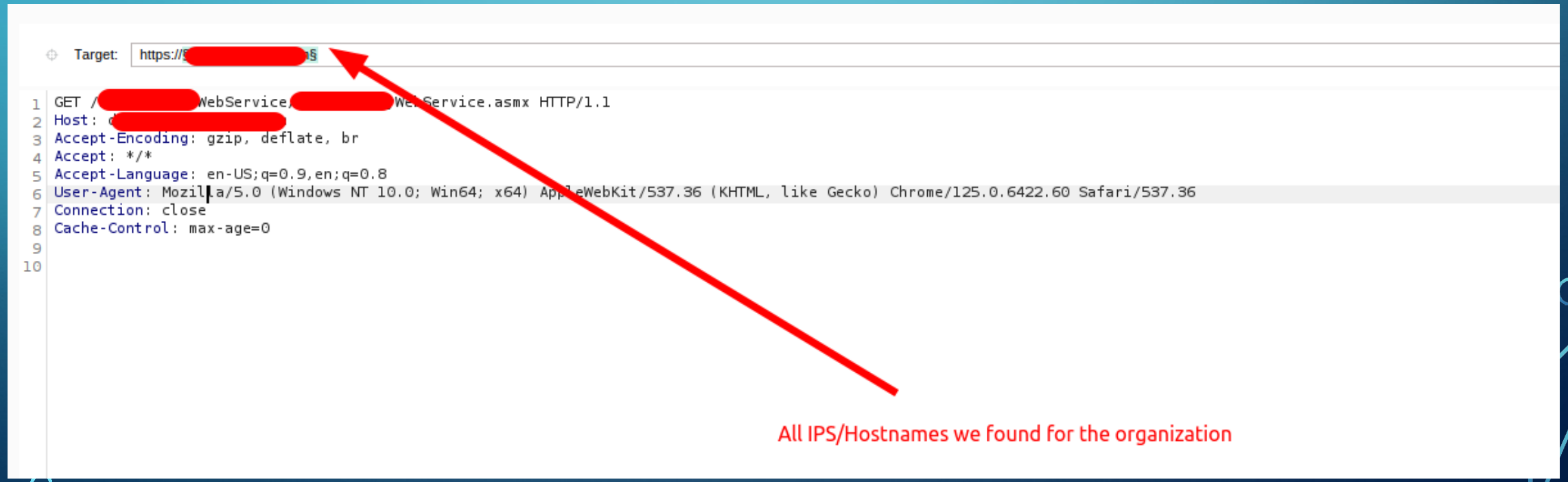
CASE STUDY

- Decompiled code showed some localhost URLs



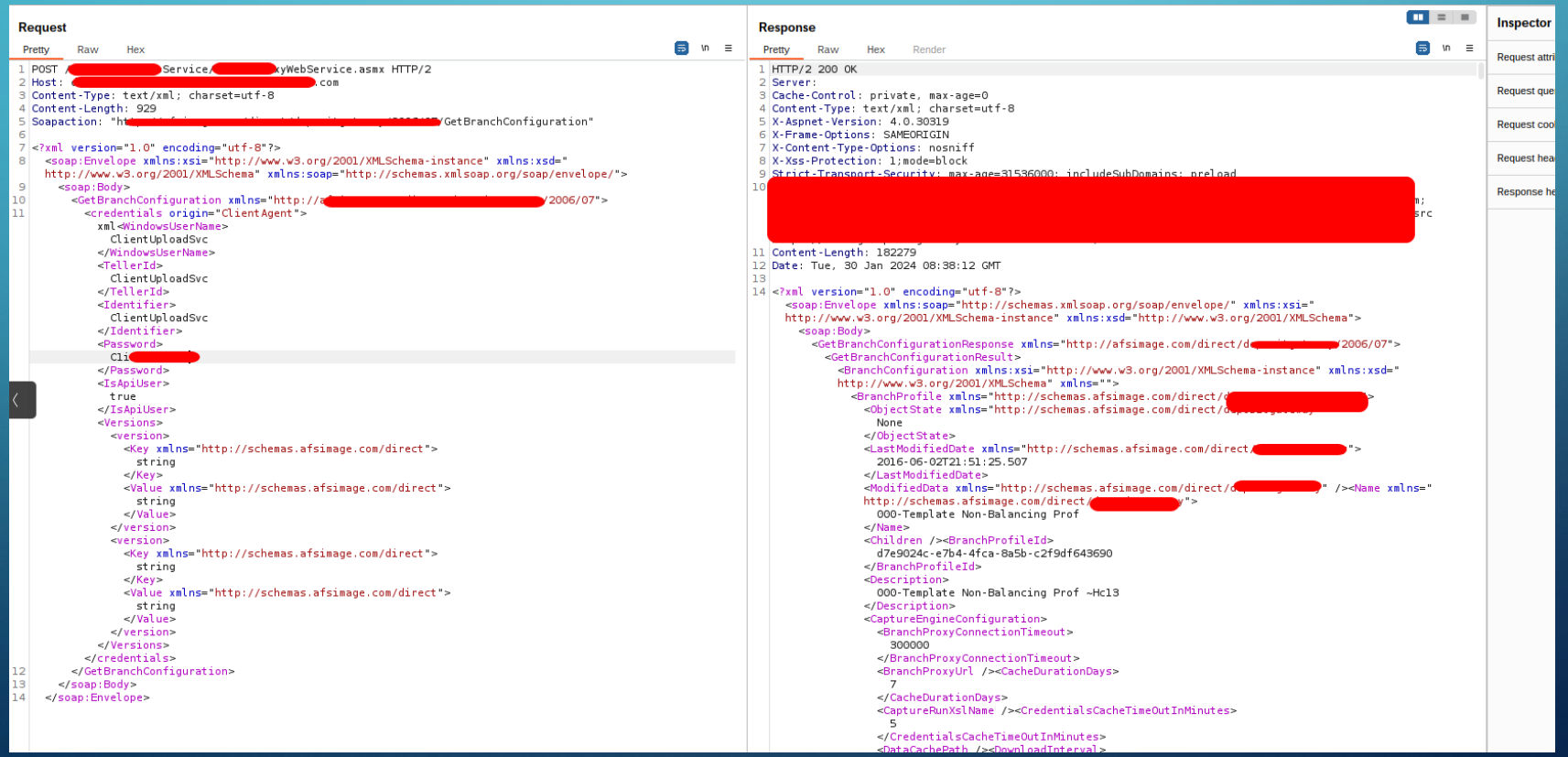
CASE STUDY

- It’s now time to find the endpoint on one of the web servers of the org – we used burp intruder



CASE STUDY

- Found a subdomain with the path we are looking for! Time to send the correct credentials and XML data



TAKEAWAYS

- Don’t be afraid to try new things
- Nothing is sure, nothing is unsure. Try what you have in mind and hope for the best.
- Spend a lot of time gathering intel about the org
- Hack when you feel like hacking

THANKS!