

# Bug Bounty - Content Discovery / Recon

Obtain subdomains and links from the target host:

```
for h in $(cat hosts.txt); do curl -sIL https://$h | egrep -io "[0-9a-z_-\.\.]+\.[0-9a-z_-\.\.]+\." | sed -e 's/https://$h/https://$h/' | sort -fu ; done
```

Obtain subdomain and V-host enumeration:

```
gobuster dns -r 8.8.8.8 --wildcard -d targetdomain.com -t 50 -c -i -w  
~/SecLists/Discovery/DNS/subdomains-top1million-20000.txt -z -q > tmp.txt && cat tmp.txt |  
cut -d' ' -f2 | sort -u > subs.txt && cat tmp.txt | cut -d' ' -f3 | tr -d '[]' | sort -u -V >  
hosts.txt
```

Obtain subdomains using Crt.sh:

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

Obtain subdomains using WebArchive:

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

Obtain subdomains using Hackertarget:

```
curl https://api.hackertarget.com/hostsearch/?q=targetdomain.com | grep -o  
'\w.*targetdomain.com'
```

Enumerate hosts from SSL certificates:

```
echo | openssl s_client -connect https://targetdomain.com:443 | openssl x509 -noout -text |  
grep DNS
```

Finding site endpoints using CommonCrawl:

```
echo "targetdomain.com" | xargs -I domain curl -s "http://index.commoncrawl.org/CC-MAIN-2018-22-index?url=*.domain&output=json" | jq -r .url | sort -u
```

Grab titles of webpages:

```
for i in $(cat Webservers.txt ); do echo "$i | $(curl --connect-timeout 0.5 $i -so - | grep -iPo '(<=<title>)(.)*(?=</title>)'"); done
```

netcat scanner for HTTP servers:

```
for i in $(seq 1 255); do nc -n -v -z "192.168.1.$i" 80 | grep "open"; done | tee webservers.txt
```

Manually perform a HTTP request:

```
# Manually perform a HTTP Get Request
echo -ne "GET / HTTP/1.0\n\n" | nc www.redspin.com 80
```

## Manually perform a HTTP Get Request on a SSL Port

```
echo -ne "GET / HTTP/1.0\n\n" | socat -- OPENSSL:www.website.com:443,verify=0
```

Create a local TCP pipe to a remote SSL port (to allow netcat to probe a SSL service):

```
socat -vd TCP-LISTEN:8888,fork OPENSSL:www.redspin.com:443,verify=0
```

Perform a check on a list of webserver (HTTP or HTTPS): HOST:PORT -> HOST:PORT | WEB SERVER | HTML Title:

```
cat webservers.txt | xargs -P10 -I'{}' bash -c '(curl -Liks -m2 "https://{ }" || curl -Liks -m2 "{ }") | grep -iao -e "^Server: .*" -e "" | sed "s#Server: \(.*\)#| \1 | #i;s###ig" | tr -d "\r\n" | sed "1s/^/{ }/;\$a\\\" | sed "s/^([^\ ]*)|$/\1|/' | tee webserver_info.txt
```

## Check if Trace is enabled on a given website:

```
echo -ne "TRACE /something HTTP/1.0\nX-Header: Trace Enabled\n\n" | socat -  
OPENSSL: www.website.com: 443, verify=0
```

## Simple HTTPS (SSL) Listener with a bad self-signed server certificate:

```
echo -ne "\n\n\n\n\n\n\n" | openssl req -new -newkey rsa:1024 -days 1 -nodes -x509 -keyout  
out.pem -out out.pem ; openssl s_server -cert out.pem -www
```

## Printing IP addresses of scope + some magic:

```
# Sort by IP Addresses  
sort -n -t. -k1,1 -k2,2 -k3,3 -k4,4
```

## Print IP addresses in a file

```
egrep -o '[:digit:]{1,3}.[:digit:]{1,3}.[:digit:]{1,3}.[:digit:]{1,3}'
```

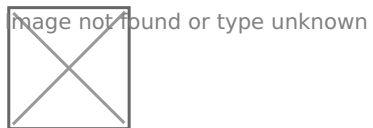
```
#Print IP address in all files in the current directory tree with some pretty color matching  
find . -type f -exec egrep -a -H -n --color=auto  
'[:digit:]{1,3}.[:digit:]{1,3}.[:digit:]{1,3}.[:digit:]{1,3}' {} ;
```

---

## By Boschko

- My Hack The Box: <https://www.hackthebox.eu/home/users/profile/37879>
- My Website: <https://olivierlaflamme.github.io/>
- My GitHub: <https://github.com/OlivierLaflamme>

- My WeChat QR below:



---

Revision #2

Created 24 September 2022 00:27:12 by mxrch

Updated 10 July 2025 16:06:33 by BlackWasp