Active Reconnaissance

Martin Stanek

2025

Table of Contents

Introduction

- Reconnaissance: passive and active
- Passive part:
  - usually a prerequisite for the active part
  - collecting information about target without interaction
- Active part:
  - fingerprinting, scanning, enumeration
  - active interaction with target systems
- Advantages:
  - more reliable and up-to-date results
  - more information obtained
- Drawbacks:
  - careless scans can be detected
  - some scans can negatively impact targets

Goals

- Goals depend on type of performed testing/assessment
  - define explicitly – avoid a fishing expedition
- Find network addresses of live hosts and network devices
- Determine OS of live hosts
- Enumerate ports and services running on the hosts
- Identify potential vulnerabilities

DNS enumeration

- brute force and dictionary enumeration
- enumeration observable on authoritative DNS servers
  - factors: rate, source (open resolvers can be used)
- list of the most frequent subdomain names
  - see the lists provided with DNS reconnaissance tools (Amass, DNSRecon) or SecLists
- reverse DNS for IP addresses – PTR records, might be missing

```
$ dig @1.1.1.1 -x 158.195.6.138 +short
www.uniba.sk.
enlight.uniba.sk.
uniba.sk.
dizajn.uniba.sk.
granty.uniba.sk.
```

- reverse IP lookup (OSINT)
  - finding A records for an IP address
  - example: identifying virtual websites served by the host
- output: list of names and IP addresses

ICMP

- Internet Control Message Protocol (ICMP)
- Message type 8: echo (ping)
  - often used for network troubleshooting
  - sometimes blocked by firewalls (especially from external networks)
  - Windows Firewall blocks ping requests by default
  - not a reliable detection of up/down hosts
- ping by nmap: `nmap www.uniba.sk -PE -sn`
  - (`-sn` no port scan)
  - other ICMP types: timestamp (`-PP`), address mask (`-PM`)

```
┌──(kali㉿kali)-[~]
└─$ nmap 158.195.6.138 -PE -sn
Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-12 15:37 EDT
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 2.06 seconds
```

ping 8.8.8.8 -c 1

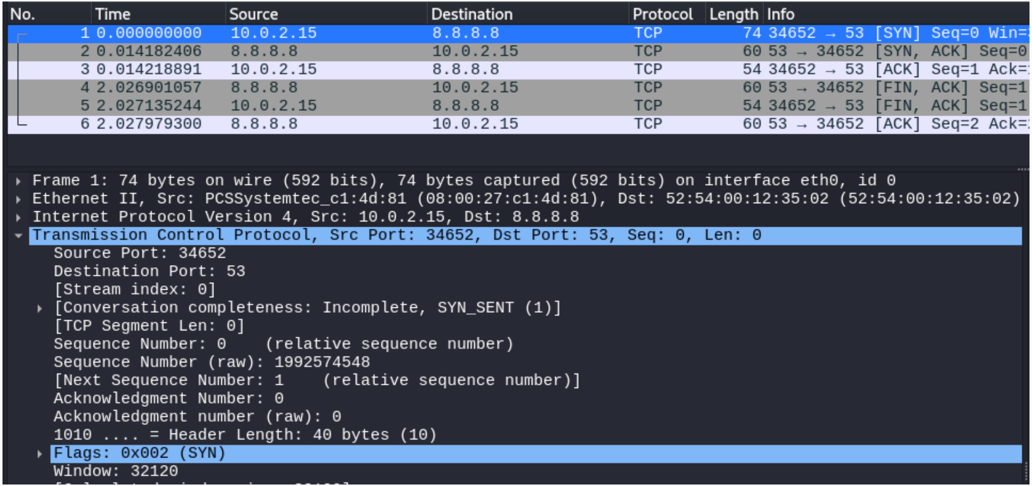
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.2.15	8.8.8.8	ICMP	98	Echo (ping) request id=0x0
2	0.009145683	8.8.8.8	10.0.2.15	ICMP	98	Echo (ping) reply id=0x0

```
▸ Frame 1: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface eth0, id 0
▸ Ethernet II, Src: PCSSystemtec_6e:13:6e (08:00:27:6e:13:6e), Dst: 52:55:0a:00:02:02 (52:55:0a:00:02:02)
▸ Internet Protocol Version 4, Src: 10.0.2.15, Dst: 8.8.8.8
▾ Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0x2605 [correct]
    [Checksum Status: Good]
    Identifier (BE): 2 (0x0002)
    Identifier (LE): 512 (0x0200)
    Sequence Number (BE): 1 (0x0001)
    Sequence Number (LE): 256 (0x0100)
    [Response frame: 2]
    Timestamp from icmp data: Mar 12, 2025 15:18:18.515903000 EDT
    [Timestamp from icmp data (relative): 0.000026102 seconds]
  ▾ Data (40 bytes)
      Data: 101112131415161718191a1b1c1d1e1f202122232425262728292a2b2c2d2e2f3031323334353637
      [Length: 40]
```

TCP

- most services use TCP
- start of a TCP connection – 3-way handshake (no data sent yet):
    1. client $\rightarrow$ server: SYN
    2. server $\rightarrow$ client: SYN/ACK
    3. client $\rightarrow$ server: ACK
- after the handshake is finished, data can be transferred

nc 8.8.8.8 53



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.2.15	8.8.8.8	TCP	74	34652 → 53 [SYN] Seq=0 Win=…
2	0.014182406	8.8.8.8	10.0.2.15	TCP	60	53 → 34652 [SYN, ACK] Seq=0…
3	0.014218891	10.0.2.15	8.8.8.8	TCP	54	34652 → 53 [ACK] Seq=1 Ack=…
4	2.026901057	8.8.8.8	10.0.2.15	TCP	60	53 → 34652 [FIN, ACK] Seq=1…
5	2.027135244	10.0.2.15	8.8.8.8	TCP	54	34652 → 53 [FIN, ACK] Seq=1…
6	2.027979300	8.8.8.8	10.0.2.15	TCP	60	53 → 34652 [ACK] Seq=2 Ack=…

```
Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface eth0, id 0
Ethernet II, Src: PCSSystemtec_c1:4d:81 (08:00:27:c1:4d:81), Dst: 52:54:00:12:35:02 (52:54:00:12:35:02)
Internet Protocol Version 4, Src: 10.0.2.15, Dst: 8.8.8.8
Transmission Control Protocol, Src Port: 34652, Dst Port: 53, Seq: 0, Len: 0
    Source Port: 34652
    Destination Port: 53
    [Stream index: 0]
    [Conversation completeness: Incomplete, SYN_SENT (1)]
    [TCP Segment Len: 0]
    Sequence Number: 0    (relative sequence number)
    Sequence Number (raw): 1992574548
    [Next Sequence Number: 1    (relative sequence number)]
    Acknowledgment Number: 0
    Acknowledgment number (raw): 0
    1010 .... = Header Length: 40 bytes (10)
    Flags: 0x002 (SYN)
    Window: 32120
```

Port scanning

- goal: detect open ports
- nmap – the most popular tool for network/port scanning
- focus on TCP and UDP scanning
- useful for network administration tasks in general

SYN scan

- SYN scan – standard scan type
  - send SYN packet
  - response SYN/ACK: port *open* (and abort handshake with RST)
  - response RST/ACK: port *closed*
  - no response: port *filtered*
- `nmap www.uniba.sk -sS -Pn -p443,666` (`-Pn` skips the host discovery stage)

```
PORT    STATE    SERVICE
443/tcp open     https
666/tcp filtered doom
```

No.	Time	Source	Destination	Protocol	Length	Info
9	5.088671640	10.0.2.15	158.195.6.138	TCP	58	39436 → 443 [SYN] Seq=0 Win
10	5.089457318	10.0.2.15	158.195.6.138	TCP	58	39436 → 666 [SYN] Seq=0 Win
11	5.093776719	158.195.6.138	10.0.2.15	TCP	60	443 → 39436 [SYN, ACK] Seq=
12	5.093830531	10.0.2.15	158.195.6.138	TCP	54	39436 → 443 [RST] Seq=1 Win
15	6.190672489	10.0.2.15	158.195.6.138	TCP	58	39438 → 666 [SYN] Seq=0 Win

Some other TCP scanning techniques

- Connection scan
  - creates a TCP connection, i.e. no privilege needed for sending raw packets
  - noisier than SYN scan, easier to spot in logs
- ACK scan (`-PA`)
  - sending a packet with just ACK flag set
  - response RST: port *unfiltered* (might be open or closed)
  - no response or some ICMP errors: *filtered*
- other techniques exist, for example:
  - Null, FIN, Xmas scans – setting none, FIN or combination of flags (URG, PSH, FIN)
  - success depends on target's implementation of TCP stack or configuration of a firewall

UDP

- connectionless (stateless) protocol
- common services that use UDP:
  - DNS (Domain Name Service, port 53)
  - DHCP (Dynamic Host Configuration Protocol, ports 67, 68)
  - NTP (Network Time Protocol, port 123)
  - SNMP (Simple Network Management Protocol, ports 161, 162)
- UDP scan
  - send UDP datagram (for known ports protocol-specific payload, empty otherwise)
  - any response: port *open*
  - no response: port *open* or *filtered*
  - ICMP error *unreachable*: port *closed*
  - other ICMP error: *filtered*

UDP example

- `nmap -sU -Pn ns.dcs.fmph.uniba.sk -p53,161,20007`

```
PORT      STATE         SERVICE
53/udp    open          domain
161/udp   open|filtered snmp
20007/udp open|filtered unknown
```

No.	Time	Source	Destination	Protocol	Length	Info
11	5.599969386	10.0.2.15	158.195.18.163	DNS	72	Standard query 0x0006 TXT version.bind
12	5.600011397	10.0.2.15	158.195.18.163	DNS	54	Server status request 0x0000
13	5.600039774	10.0.2.15	158.195.18.163	UDP	42	34505 → 20007 Len=0
14	5.600069374	10.0.2.15	158.195.18.163	SNMP	93	get-request 1.3.6.1.2.1.1.5.0
15	5.600100738	10.0.2.15	158.195.18.163	SNMP	102	get-request
16	5.606647776	158.195.18.163	10.0.2.15	DNS	104	Standard query response 0x0006 TXT versi

Scanning performance

- scanning subnets
  - example: Comenius University
  - IP range 158.195.0.0 - 158.195.255.255, $\approx 2^{16}$ addresses
- scanning all ports ($2^{16} - 1$ TCP ports, the same number of UDP ports)
- too many probes
  - slow, e.g. nmap default RTT (Round Trip Timeout) is 10 seconds
  - noisy – one or more packets for each host/port combination
  - possibility to trigger IDS, firewall rules
- some strategies (we discuss them on the following slides):
  1. Sample subset of IP addresses
  2. Subset of TCP/UDP ports
  3. Analyze firewall rules to narrow the scope
  4. Fast scanning methods

Sample subset of IP addresses

- sampling from the set of all web servers, desktop PCs, notebooks, etc.
- suitable for common/standardized configurations
- sample size matters
- easy to miss something
  - statistical testing is not suitable for finding a singular vulnerability

Subset of TCP/UDP ports

- testing the most frequent ports (top 100, 1000 etc.)
- often used for host discovery
  example: `nmap -sn -PE -PS80,443,3389 -PP -PU161,40125 -PA21 --source-port 53`
- unknown status of untested ports
- top 12 TCP ports (will be different for particular target):

#	port	description	#	port	description	#	port	description
1.	80	HTTP	5.	22	SSH	9.	445	SMB
2.	23	telnet	6.	25	SMTP	10.	139	NETBIOS
3.	443	HTTPS	7.	3389	MS RDP	11.	143	IMAP
4.	21	FTP	8.	110	POP3	12.	53	DNS

- top 12 UDP ports:

#	port	description	#	port	description	#	port	description
1.	631	IPP	5.	138	NETBIOS	9.	67	DHCP
2.	161	SNMP	6.	1434	MS SQL	10.	53	DNS
3.	137	NETBIOS	7.	445	SMB	11.	139	NETBIOS
4.	123	NTP	8.	135	MS RPC	12.	500	ISAKMP

Analyze firewall rules to narrows the scope

- efficient, especially for large network ranges
- requires a cooperation or firewall access
- not black-box anymore, partially a configuration review
- might be more work than expected (number of rules)
- combine with full scan (all ports) for sample targets
  - verify that firewall rules work as expected

Fast scanning methods

- strategies:
  - paralel scanning with multiple machines
  - fast rate for sending packets
- tools: zmap, masscan
  - asynchronous: separate threads for sending and receiving packets
  - less accurate – some open ports can remain undetected
  - bandwidth sensitive – huge number of transmitted packets
- zmap
  - single port testing for large number of targets
- massscan
  - custom TCP/IP stack, custom network driver (if needed)
  - randomizes the target IP addresses (prevents DoS)

Remarks

- DNS used for load balancing/failover
    - potentially scanning multiple hosts, merged results
    - IP address is more reliable target (although it can be distributed among multiple physical systems too)

Network tracing

- tracing communication path – information about network architecture
- limit how many hops can packet/datagram make (counter)
  - IPv4 header: time to live (TTL)
  - IPv6 header: hop limit
  - each router should decrease the counter
  - if counter is 0: discard the packet and return ICMP message to the source address
    - ICMPv4: type 11, TTL exceeded in transit
    - ICMPv6: type 3, Hop limit exceeded in transit
- traceroute/tracert
  - find routers along the path from the source to target IP address
  - sending packets (TCP, UDP, ICMP, specific destination port, specific flags, etc.):
  - start with TTL 1 and gradually increase the counter

Network tracing – web services and limits

- various web services for traceroute
  - different source machines, i.e. different results
- limits:
  - some routers do not send ICMP reply or block the probe (*)
  - load balancers – replies from different paths (unreliable output)
  - Paris traceroute, Dublin traceroute etc. – Multi-path Detection Algorithm (MDA)

Traceroute example

```
Tracing route to snm.sk [93.184.69.234]
over a maximum of 30 hops:

  1    <1 ms    <1 ms    <1 ms  192.168.100.1
  2    18 ms     8 ms     9 ms  [redacted].dynamic.orange.sk [[redacted]]
  3     3 ms     3 ms     3 ms  192.168.115.25
  4     3 ms     3 ms     2 ms  213-151-198-26.static.orange.sk [213.151.198.26]
  5     3 ms     3 ms     3 ms  Vnet-gw1.six.sk [192.108.148.210]
  6     4 ms     3 ms     4 ms  hu31.sixncs2.noc.vnet.sk [185.176.73.238]
  7     3 ms     3 ms     3 ms  hu24.shc3ncs2.noc.vnet.sk [185.176.73.227]
  8     3 ms     3 ms     3 ms  eth1-2.n931.noc.vnet.sk [185.176.73.193]
  9     3 ms     3 ms     3 ms  customer4.elet.sk [93.184.69.234]
```

8	GW-VNet.retn.net 87.245.246.115	RETN-AS, GB	87.245.224.0/19
9	vl595.n9sit2.vnet.sk 185.176.72.71	VNET-AS, SK	185.176.72.0/22
10	po932.n932.vnet.sk 185.176.72.89	VNET-AS, SK	185.176.72.0/22
11	customer4.elet.sk 93.184.69.234	VNET-AS, SK	93.184.64.0/20

ARP scan

- ARP (Address Resolution Protocol)
  - layer 2 protocol
  - usually maps IP address to MAC address
    1. request – broadcast: destination IP, source IP and MAC
    2. response – unicast:
  - local ARP cache of learned IP and MAC pairs
  - MAC address – vendor identification
  - assumptions: IPv4 hosts, local network (non-routable)
  - IPv6 uses Neighbor Discovery Protocol (NDP) instead of ARP
- ARP scan
  - send request for (a subset of) all IP addresses in local network
  - passive approach: just listen for ARP broadcasts (any sniffer can do)
  - tools: arp-scan, netdiscover, nmap

ARP scan example

- fragment of arp-scan for local network (158.195.87.0/25)

```
# arp-scan --localnet
Interface: enp0s25, type: EN10MB, MAC: 18:03:73:c1:16:a3, IPv4: 158.195.87.21
Starting arp-scan 1.9.7 with 128 hosts (https://github.com/royhills/arp-scan)
158.195.87.9    00:0e:0c:4e:05:54       Intel Corporation
158.195.87.30   d8:cb:8a:b1:f4:92       Micro-Star INTL CO., LTD.
158.195.87.31   00:0b:82:3a:11:44       Grandstream Networks, Inc.
158.195.87.34   10:bf:48:b5:5e:5d       ASUSTek COMPUTER INC.
158.195.87.39   18:03:73:c1:16:89       Dell Inc.
158.195.87.44   d8:cb:8a:b1:f4:a6       Micro-Star INTL CO., LTD.
158.195.87.78   98:90:96:c0:3f:b6       Dell Inc.
158.195.87.82   98:90:96:c0:3d:bd       Dell Inc.
158.195.87.83   b8:ac:6f:23:ca:94       Dell Inc.
158.195.87.88   10:7b:44:4a:86:aa       ASUSTek COMPUTER INC.
158.195.87.96   9c:7b:ef:82:ca:b8       Hewlett Packard
```

OS identification

- fingerprinting the target
  - decide on further steps (vulnerabilities)
  - info on unknown devices
- send specific packets and observe behavior
- examples of test preformed on responses (used by nmap OS fingerprinting):
  - TCP ISN (initial sequence number) greatest common divisor
  - TCP ISN counter rate
  - TCP options (what options are set and their order)
  - TCP initial windows size
  - IP initial TTL
- often just general info (Linux, Windows, VoIP phone, etc.), version unreliable
- service enumeration and version detection may help with OS identification

OS identification examples

- selected detection results (`nmap -O <IP address>`):
- Ubuntu 20.04 (localhost, kernel 5.4.0-42-generic):

  ```
  Device type: general purpose
  OS details: Linux 2.6.32
  ```
- VoIP phone (Grandstream GXP2000):

  ```
  Aggressive OS guesses: Grandstream GXP2000 VoIP phone (95%), Revo
  Blik Wi-Fi Internet radio (94%), Rigol DSG3060 signal generator
  (93%), ...
  No exact OS matches for host (test conditions non-ideal).
  ```

Services

- ports are identified, next: what service is listening?
  - port scan shows a standard service name
    (IANA, Service Name and Transport Protocol Port Number Registry)
  - nmap: `nmap-services` file
  - services listening on non-standard ports
  - services moved to other ports
- specific probes
  - nmap: `nmap-service-probes` file
  - list of probes used for detection
- nmap scripts (NSE – nmap scripting engine)
  - `-sC` default set of scripts
  - categories: auth, broadcast, brute, default, discovery, dos, exploit, external, fuzzer, intrusive, malware, safe, version, vuln

Services enumeration example 1

- selected detection results (`nmap -sV <IP address>`)
- Windows 10, TCP ports:

```
PORT      STATE SERVICE VERSION
135/tcp   open  msrpc   Microsoft Windows RPC
445/tcp   open  microsoft-ds?
2869/tcp  open  http    Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
16992/tcp open  http    Intel Active Management Technology User
                        Notification Service httpd 10.0.60
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows,
              cpe:/h:intel:active_management_technology:10.0.60
```

Services enumeration example 2

- VoIP phone (Grandstream GXP2000), TCP ports:

```
PORT   STATE SERVICE VERSION
23/tcp open  telnet  Grandstream GXP2000 VoIP phone telnetd
80/tcp open  http    Grandstream GXP2000 http config 1.1.4.18
Service Info: Devices: VoIP phone, VoIP adapter;
              CPE: cpe:/h:grandstream:gxp2000
```

Other objects for enumeration

- usual assumption: a presence in the system, on local network, etc.
- users, computers, groups
  - LDAP, Active Directory searches (users, groups, computers, etc.)
  - web application user enumeration (considered an authentication weakness)
- web application directory enumeration (brute-force, crawling)
- processes
- installed applications
- specific tools for specific frameworks and IT infrastructure, e.g.
  - WordPress (WPScan), containers and their orchestration, cloud platforms

Web applications – relevant information

- technology stack
  - directing further testing (efficiency)
  - vulnerable versions
  - HTTP headers, Cookies, source code, specific files, error messages, file extensions
  - tools: WhatWeb, Wappalyzer, BuiltWith
- directories/files discovery (forced browsing)
  - tools: gobuster, feroxbuster, ffuf
- source code (web pages, javascript)
  - passwords, comments
- archives
  - what is changed (old parts can be still present)
  - vulnerabilities introduced, not fixed or not fixed thoroughly

WhatWeb example (www.uniba.sk)

```
└─$ whatweb www.uniba.sk
http://www.uniba.sk [301 Moved Permanently] Apache[2.2.22], Country[S
LOVAKIA (Slovak Republic)][SK], HTTPServer[Debian Linux][Apache/2.2.2
2 (Debian)], IP[158.195.6.138], RedirectLocation[https://uniba.sk/],
Title[301 Moved Permanently]
https://uniba.sk/ [200 OK] Apache[2.2.22], Bootstrap, Cookies[fe_typo
_user], Country[SLOVAKIA (Slovak Republic)][SK], Email[infocentrum@un
iba.sk], Frame, HTML5, HTTPServer[Debian Linux][Apache/2.2.22 (Debian
)], IP[158.195.6.138], JQuery[1.10.1], MetaGenerator[TYPO3 4.7 CMS],
Open-Graph-Protocol, PoweredBy[TYPO3], Script[javascript,text/javascr
ipt], TYPO3[4.7], Title[Univerzita Komenského], X-UA-Compatible[IE=ed
ge], YouTube
```

Vulnerabilities

- automatic scanning of known vulnerabilities in 3rd party software
  - Nessus (Tenable), Qualys, OpenVAS, etc.
  - capable to perform the host discovery and enumeration as well
  - usually produces an intense network communication
- Vulnerability management (internal IT process)
  - testing the security of your infrastructure
  - *a separate lecture*
- penetration testing usually more selective and focused
  - exploitable vulnerabilities (based on enumeration of services, versions)
  - known credentials (or dictionary attacks)
  - application vulnerabilities (e.g. web apps)
  - specific tools, such as Metasploit framework (integrates various tools and stages of a pentest), Burp suite, etc.

Exploits

- specific modules in a framework, such as Metasploit
- public exploit database: exploit-db.com
- just search for an exploit
- develop your own (rarely)
- show that it could be exploited
    - (somewhat) realistic scenario

Exercises

1. TryHackMe: Vulnversity
   - discuss how you did the privilege escalation task
   - get the password hash of the only non-root user (screenshot `/etc/shadow`)
2. Collect ARP information on your local network
   - screenshot, redact sensitive information
   - assign/find vendors for reported MAC addresses
   - learn how to change a MAC address in your operating system (verify)

Additional resources

1. Nmap Reference Guide
2. R. Davis, *The Art of Network Penetration Testing*, Manning Publications, 2020
3. Carlos Polop, *HackTricks*