ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
UNIVERSITY OF PIRAEUS

CDS201: Έλεγχος Εισβολών Δικτύων και Συστημάτων (Penetration Testing)

Presentation Outline

- External Recon – OSINT tools
- Information Gathering – Active/Passive
- Network Enumeration with Nmap
- HTB labs

External Reconnaissance

External reconnaissance and information gathering are crucial components of penetration testing for:

- **Understanding the Attack Surface**: identify the various entry points and vulnerabilities. By comprehensively mapping out the attack surface, pentesters can better understand potential avenues for exploitation.
- **Identifying Weaknesses**: identify weaknesses in the external infrastructure (misconfigured servers, outdated software, or publicly accessible sensitive information). These weaknesses can be exploited by real threat actors to gain unauthorized access.
- **Assessing Security Posture**: assess the effectiveness of the security controls from an outsider's perspective. This helps in identifying gaps in the security posture that need to be addressed to enhance overall resilience against cyber threats.
- **Real-world Attack Simulation:** simulating attacks that adversaries might launch against the target organization. Adopting the mindset and techniques of potential attackers, pentesters can provide valuable insights into the organization's readiness to defend against such threats.
- **Prioritizing Targets:** Information gathered during external recon helps pentesters to prioritize their targets based on the level of risk they pose to the organization. Then, testing efforts on the most critical assets and vulnerabilities, maximizing the impact of the pentest.
- **Customizing Attack Strategies**: provides valuable intelligence that can be used to customize attack strategies tailored to the target organization's specific environment. This increases the likelihood of successfully exploiting vulnerabilities and gaining unauthorized access.

External Reconnaissance

- In case of a not “assume breached” scenario/engagement (an initial access is not provided)
- A vital phase because it provides information that can be used:
  - To exploit the target
  - Or gain access to data
- 2 main facets of recon - **organizational** and **technical**
- **Organizational:** collecting information about the organization
  - people who work there (names, jobs and skills)
  - organizational structure
  - site locations and business relationships
- **Technical:** looking for systems
  - public-facing websites
  - mail servers
  - remote access solutions
  - defensive vendors or programs in use: web proxies, email gateways, firewalls, antivirus etc.

External Reconnaissance

Information gathering **Passively** or **Actively**

Passive: not actively touching parts of the target network

- Google, LinkedIn, Shodan and social media

Active: directly touching those components

- visiting the target's website
- port scanning their IP ranges
- **Riskier**: provides an organization with their first potential indication that they're being looked at
- **doing so via a proxy or VPN service to not expose public IP addresses**

External Reconnaissance

DNS Records

- Domain Name System (DNS) records can provide a lot of information regarding services that may be exposed to the Internet.

```
$ dig targetdomain.com
$ whois  <targetIPs>
```

- Watch out for 3rd Party Cloud Providers (cloudflare, AWS, Azure)

- Subdomains can also provide insight to other publicly available services (webmail, remote access solutions such as Citrix, or a VPN)
  - Tools such as dnscan come with lists of popular subdomains
- Weak email security (SPF, DMARC and DKIM) may allow us to spoof emails to appear as though they're coming from their own domain.
  - Spoofcheck: Python tool that can verify the email security of a given domain.

External Reconnaissance

Google Dorks

- Google "dorking" is a way of using Google's advanced search operators to find very targeted information.
  - Google Hacking Database contains hundreds of examples.

  Examples:
    - **site:** Limit the search results to a specific website.
      site:apple.com
    - **intitle:** Find pages with a certain word in the title.
      intitle:apple
    - **inurl:** Find pages with a certain word in the URL.
      inurl:apple
    - **intext:** Find pages containing a certain word (or words) somewhere in the content.
      intext:apple
    - **iletype:** Search for filetypes that Google understands.
      site:apple.com filetype:pdf
    - **#..#:** Search for a range of numbers.
      site:apple.com filetype:pdf 2020..2022
    - **-:** Exclude a phrase.
      site:apple.com -www -support

External Reconnaissance

DNS Records

- Domain Name System (DNS) records can provide a lot of information regarding services that may be exposed to the Internet.

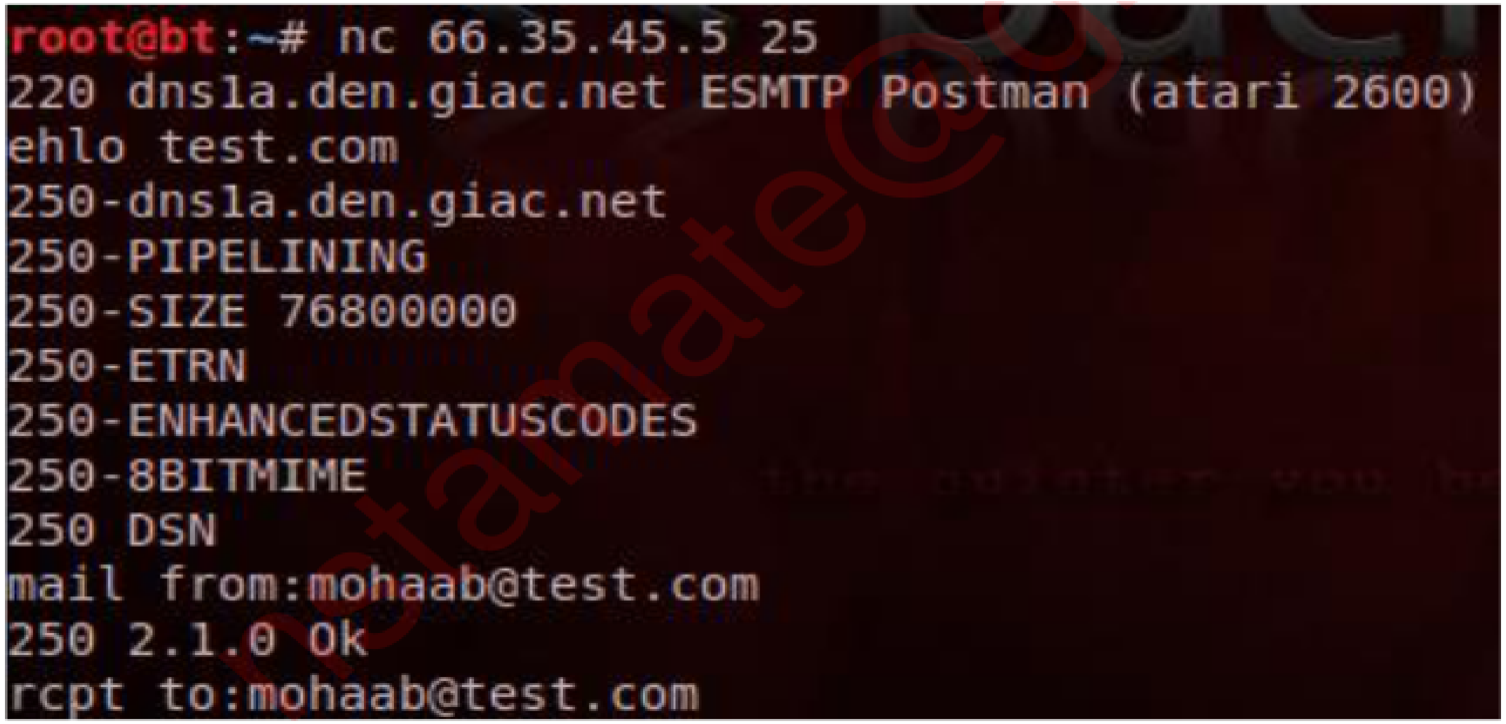
```
$ dig targetdomain.com
$ whois  <targetIPs>
```

- Watch out for 3rd Party Cloud Providers (cloudflare, AWS, Azure)

- Subdomains can also provide insight to other publicly available services (webmail, remote access solutions such as Citrix, or a VPN)
  - Tools such as dnscan come with lists of popular subdomains
- Weak email security (SPF, DMARC and DKIM) may allow us to spoof emails to appear as though they're coming from their own domain.
  - Spoofcheck: Python tool that can verify the email security of a given domain.

External Reconnaissance

Enumerating email accounts with nc manually – SMTP



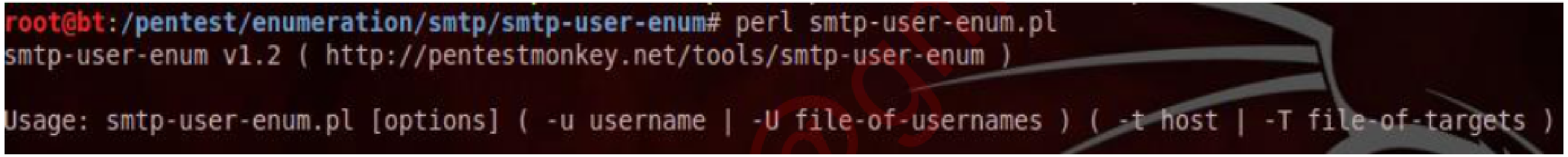
```
root@bt:~# nc 66.35.45.5 25
220 dns1a.den.giac.net ESMTP Postman (atari 2600)
ehlo test.com
250-dns1a.den.giac.net
250-PIPELINING
250-SIZE 76800000
250-ETRN
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN
mail from:mohaab@test.com
250 2.1.0 Ok
rcpt to:mohaab@test.com
```

External Reconnaissance

Enumerating email accounts with a script – SMTP

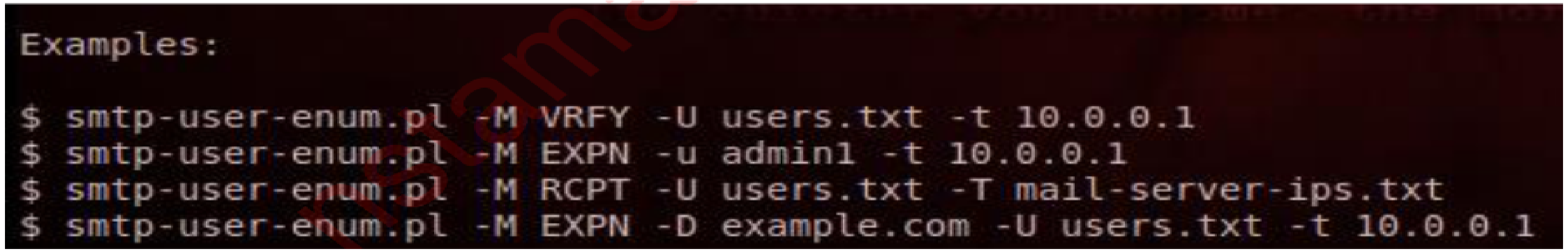
http://pentestmonkey.net/tools/user-enumeration/smtp-user-enum

$ smtp-user-enum -M VRFY -u admin -t 192.168.1.64
$ smtp-user-enum -M VRFY -U /tmp/users.txt -t 192.168.1.64



```
root@bt:/pentest/enumeration/smtp/smtp-user-enum# perl smtp-user-enum.pl
smtp-user-enum v1.2 ( http://pentestmonkey.net/tools/smtp-user-enum )

Usage: smtp-user-enum.pl [options] ( -u username | -U file-of-usernames ) ( -t host | -T file-of-targets )
```



```
Examples:

$ smtp-user-enum.pl -M VRFY -U users.txt -t 10.0.0.1
$ smtp-user-enum.pl -M EXPN -u admin1 -t 10.0.0.1
$ smtp-user-enum.pl -M RCPT -U users.txt -T mail-server-ips.txt
$ smtp-user-enum.pl -M EXPN -D example.com -U users.txt -t 10.0.0.1
```

External Reconnaissance

Social Media

- Social Media platforms such as LinkedIn, Facebook and Twitter can be a goldmine of information. This is useful for getting insight into the possible technology stacks and business processes being used.
- Many people also cross-link their social media profiles, so you can find their Twitter/Facebook/Instagram/etc accounts as well. Phishing is still the most prevalent method of compromising a target and gathering both professional and personal information on targets goes a long way to making those pre-texts convincing and enticing.
- LinkedInt automated scraping tools . However, in the case of LinkedIn, they often violate their user agreements, leading to your account being banned. If you have to use an account for scraping purposes, make sure it's a "burner".
- hunter.io can be used to discover the email address of employees. it tells us that the most common pattern for a given domain ({f}{last}@domain.com) . This means that we don't actually have to find everybody's email address explicitly, but simply guess based on this pattern. We could scrape a list of the domain employees from LinkedIn and transform their names into email address.

  For instance, Steve Jobs would become s.jobs@apple.com

External Reconnaissance

DNS Records

- Domain Name System (DNS) records can provide a lot of information regarding services that may be exposed to the Internet.

```
$ dig targetdomain.com
$ whois  <targetIPs>
```

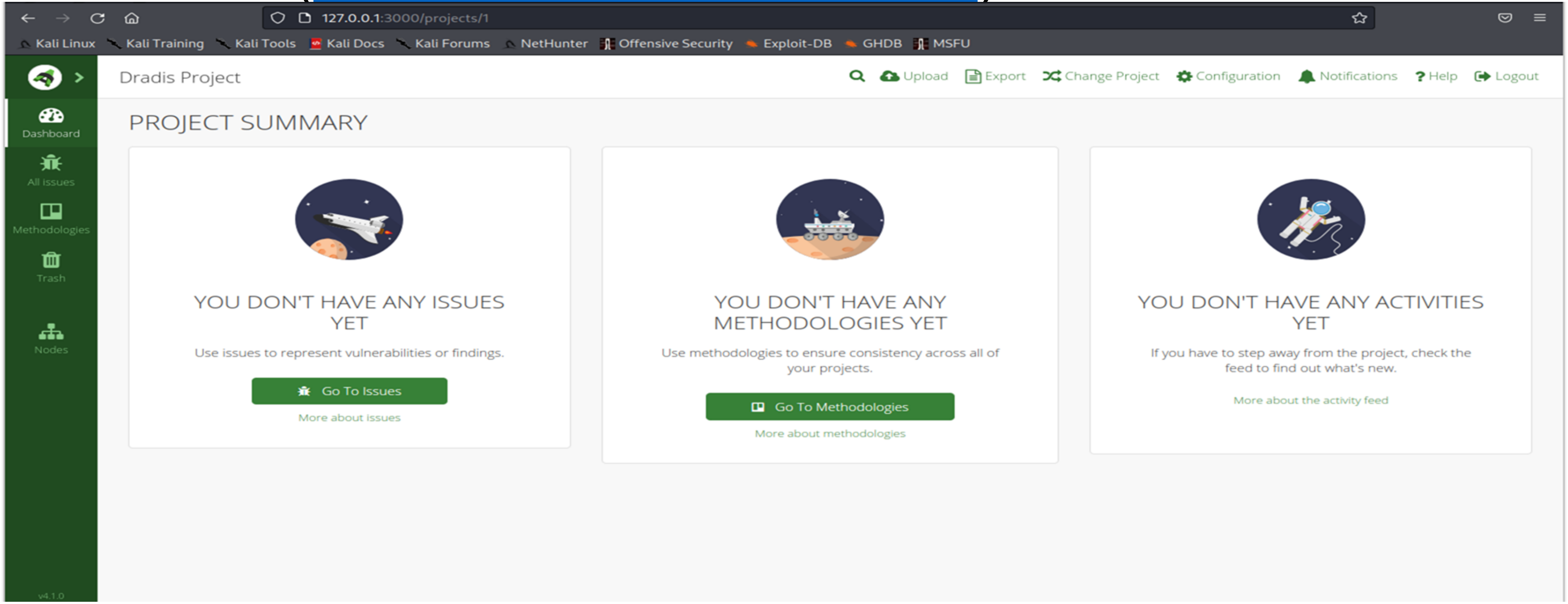
- Watch out for 3rd Party Cloud Providers (cloudflare, AWS, Azure)

- Subdomains can also provide insight to other publicly available services (webmail, remote access solutions such as Citrix, or a VPN)
  - Tools such as dnscan come with lists of popular subdomains
- Weak email security (SPF, DMARC and DKIM) may allow us to spoof emails to appear as though they're coming from their own domain.
  - Spoofcheck: Python tool that can verify the email security of a given domain.

External Reconnaissance

Organize information collected

- Dradis tool (http://dradisframework.org/)



PROJECT SUMMARY

YOU DON'T HAVE ANY ISSUES YET

Use issues to represent vulnerabilities or findings.

YOU DON'T HAVE ANY METHODOLOGIES YET

Use methodologies to ensure consistency across all of your projects.

YOU DON'T HAVE ANY ACTIVITIES YET

If you have to step away from the project, check the feed to find out what's new.

External Reconnaissance

Organize information collected

- CTFPad (https://github.com/StratumAuhuur/CTFPad)
- Dradis tool (http://dradisframework.org/)
- Magictree (http://www.gremwell.com/download)
- Gobby (apt-get install gobby)
- Lair (https://github.com/lair-framework/lair)
- Mediawiki (https://www.mediawiki.org/wiki/Download)
- ZIM (https://zim-wiki.org/downloads.html)

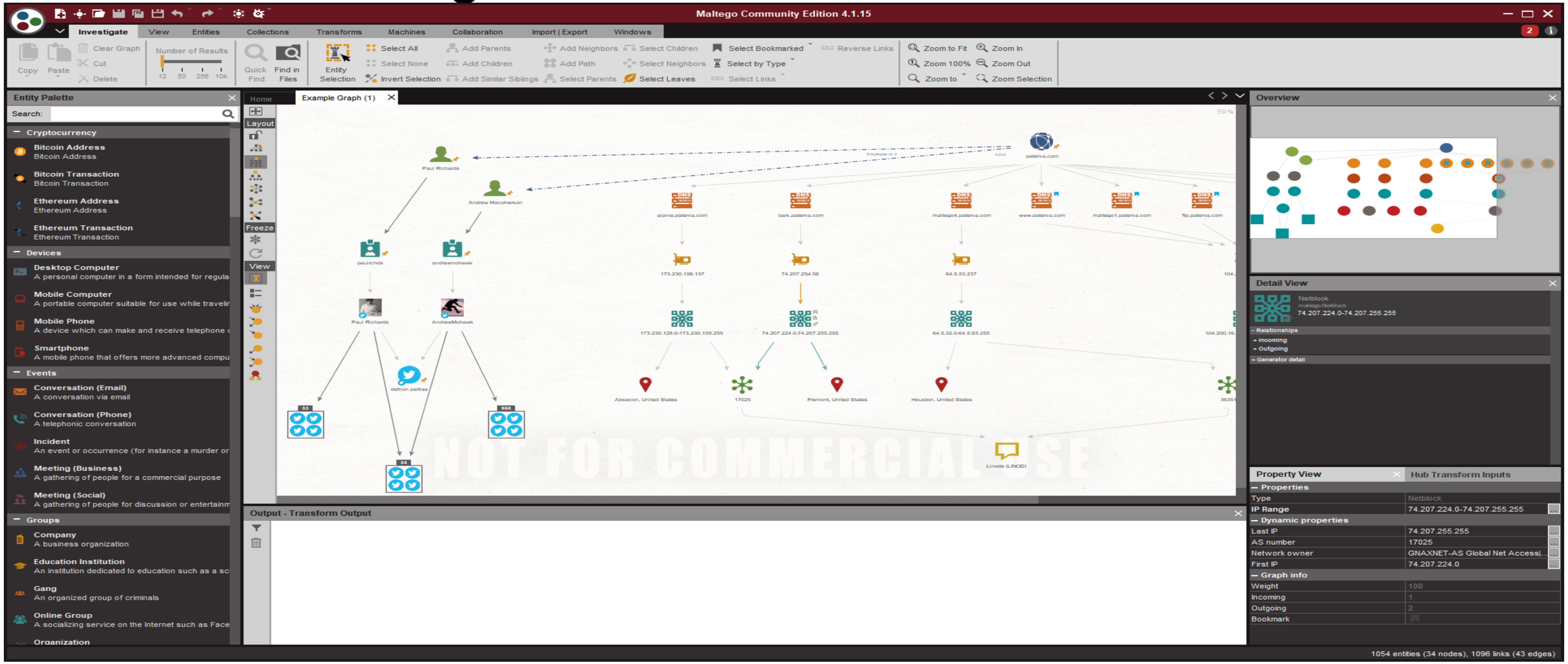
External Reconnaissance

OSINT tools

- recon-ng : a full-featured reconnaissance framework designed with the goal of providing a powerful environment to conduct open source web-based reconnaissance quickly and thoroughly
- Eyewitness: designed to take screenshots of websites provide some server header info, and identify default credentials if known
- Maltego: an open source intelligence and graphical link analysis tool for gathering and connecting information for investigative tasks. Maltego is a Java application that runs on Windows, Mac and Linux.

External Reconnaissance

OSINT tools -Maltego



Properties	
Type	Netblock
IP Range	74.207.224.0-74.207.255.255
Dynamic properties	
Last IP	74.207.255.255
AS number	17025
Network owner	GNAXNET-AS Global Net Access|...
First IP	74.207.224.0
Graph info	
Weight	100
Incoming	1
Outgoing	2
Bookmark	

1054 entities (34 nodes), 1096 links (43 edges)

Information Gathering
dig & nslookup

- Both dig and nslookup are command-line utilities used for querying Domain Name System (DNS) servers to obtain information about domain names, IP addresses, and other DNS records. In a penetration testing context, these tools can be invaluable for reconnaissance, enumeration, and identifying potential attack vectors.

Information Gathering
dig & nslookup

dig and **nslookup** commands results can be useful:

1. DNS Enumeration:

Enumerating DNS Records: Use dig or nslookup to enumerate DNS records such as A (IPv4 address), AAAA (IPv6 address), MX (mail exchange), NS (name server), CNAME (canonical name), and TXT (text) records.

Discovering Subdomains: By querying DNS records, you can discover subdomains associated with the target domain, potentially identifying overlooked attack surfaces.

Information Gathering
dig & nslookup

2. Information Gathering:

IP Address Information: Obtain IP addresses associated with domain names, which can reveal server infrastructure and hosting providers.

Mail Server Information: Identify mail exchange (MX) records to understand the email infrastructure, potentially uncovering email servers that could be targeted for phishing attacks.

3. DNS Zone Transfer:

Identifying Misconfigurations: Attempt DNS zone transfers using dig (AXFR query) or nslookup (ls command) to determine if the DNS server is misconfigured, potentially leaking sensitive information such as internal domain names and IP addresses.

Discovering Additional Domains: Zone transfers may reveal additional domains hosted on the same DNS server, expanding the attack surface.

4

Information Gathering
dig & nslookup

4. Malware Analysis and Tracking:

Identifying Malicious Domains: Use dig or nslookup to query DNS records associated with potentially malicious domains obtained from threat intelligence feeds or suspicious URLs. This can aid in malware analysis and tracking.

5. Network Reconnaissance:

Mapping Network Infrastructure: DNS information obtained through dig and nslookup can assist in mapping the target organization's network infrastructure, including servers, services, and their relationships.

6. Verifying DNS Security Configurations:

Checking DNSSEC: Use dig to verify if DNS Security Extensions (DNSSEC) are implemented on the target domain, ensuring data integrity and authentication of DNS responses.

Validating SPF and DKIM Records: Query TXT records to verify the implementation of Sender Policy Framework (SPF) and DomainKeys Identified Mail (DKIM), which enhance email security.

Information Gathering
dig & nslookup

7. Identifying Load Balancers and Redirections:

Detecting Load Balancers: Querying DNS records may reveal the presence of load balancers or multiple IP addresses associated with a single domain, indicating load balancing or failover configurations.

Analyzing Redirects: Identify DNS records associated with redirects (CNAME or A records) that could be used to bypass security controls or manipulate traffic flow.

8. Investigating Domain Ownership:

WHOIS Information: Obtain WHOIS information for domain names using external services or dig to investigate domain ownership, registration dates, and contact information.

9. Monitoring DNS Changes:

Tracking Changes: Regularly query DNS records to monitor for unauthorized changes, such as DNS hijacking or domain spoofing attempts.

Information Gathering
dig & nslookup

Examples:

Query A records:

```
$ nslookup $TARGET
$ dig facebook.com @1.1.1.1  (specifying the nameserver)
$ nslookup -query=A <www.target.com> (specify the subdomain)
$ dig a www.target.com @1.1.1.1 (specify the subdomain)
```

Information Gathering
dig & nslookup

We can query for any resource record or all of them.

Name Servers (NS)	The distributed database is bound together by NS Records. They are in charge of a zone's authoritative name server and the authority for a child zone to a name server.
IPv4 Addresses (A)	The A record is only a mapping between a hostname and an IP address. 'Forward' zones are those with A records.
Pointer (PTR)	The PTR record is a mapping between an IP address and a hostname. 'Reverse' zones are those that have PTR records.
Canonical Name (CNAME)	An alias hostname is mapped to an A record hostname using the CNAME record.
Mail Exchange (MX)	The MX record identifies a host that will accept emails for a specific host. A priority value has been assigned to the specified host. Multiple MX records can exist on the same host, and a prioritized list is made consisting of the records for a specific host.

Information Gathering
Subdomain Enumeration - Passive

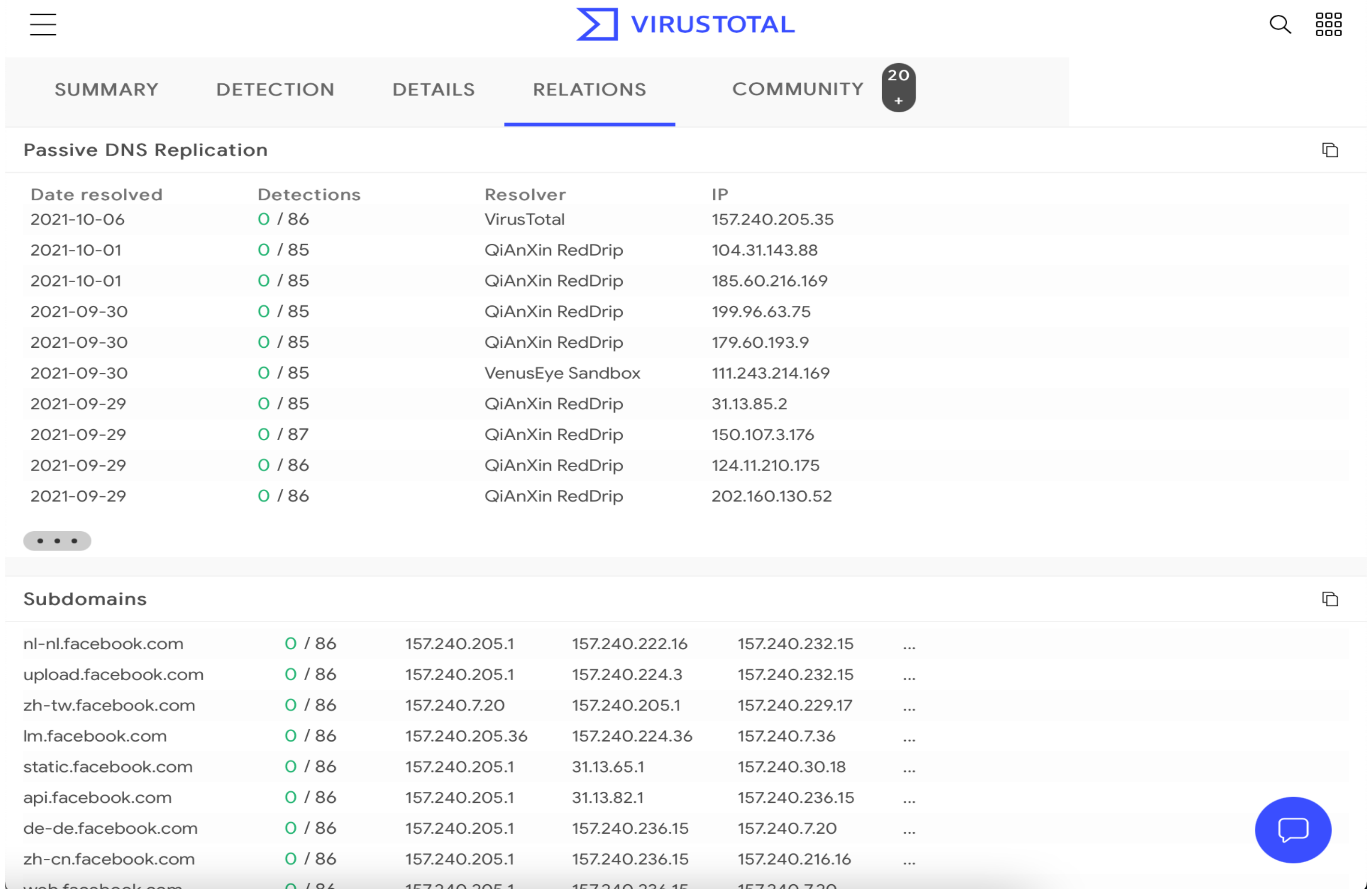
- *Goal:* mapping all available subdomains within a domain name (In scope)
- Increases the attack surface
- May uncover backend panels or intranet applications thought to be hidden in a “security by obscurity” manner
- passive subdomain enumeration using third-party services or publicly available information
- Information gathered here will be used in later active enumeration activities

Information Gathering
Subdomain Enumeration - Passive

__VirusTotal__

When users visit

Urls it preserves DNS resolutions



Passive DNS Replication

Date resolved	Detections	Resolver	IP
2021-10-06	0 / 86	VirusTotal	157.240.205.35
2021-10-01	0 / 85	QiAnXin RedDrip	104.31.143.88
2021-10-01	0 / 85	QiAnXin RedDrip	185.60.216.169
2021-09-30	0 / 85	QiAnXin RedDrip	199.96.63.75
2021-09-30	0 / 85	QiAnXin RedDrip	179.60.193.9
2021-09-30	0 / 85	VenusEye Sandbox	111.243.214.169
2021-09-29	0 / 85	QiAnXin RedDrip	31.13.85.2
2021-09-29	0 / 87	QiAnXin RedDrip	150.107.3.176
2021-09-29	0 / 86	QiAnXin RedDrip	124.11.210.175
2021-09-29	0 / 86	QiAnXin RedDrip	202.160.130.52

• • •

Subdomains

nl-nl.facebook.com	0 / 86	157.240.205.1	157.240.222.16	157.240.232.15	...
upload.facebook.com	0 / 86	157.240.205.1	157.240.224.3	157.240.232.15	...
zh-tw.facebook.com	0 / 86	157.240.7.20	157.240.205.1	157.240.229.17	...
lm.facebook.com	0 / 86	157.240.205.36	157.240.224.36	157.240.7.36	...
static.facebook.com	0 / 86	157.240.205.1	31.13.65.1	157.240.30.18	...
api.facebook.com	0 / 86	157.240.205.1	31.13.82.1	157.240.236.15	...
de-de.facebook.com	0 / 86	157.240.205.1	157.240.236.15	157.240.7.20	...
zh-cn.facebook.com	0 / 86	157.240.205.1	157.240.236.15	157.240.216.16	...
web.facebook.com	0 / 86	157.240.205.1	157.240.236.15	157.240.7.20	

Information Gathering
Subdomain Enumeration - Passive

Certificates

- *a useful source for subdomain information in SSL/TLS certificates*
- *CT (certificate transparency project): requires every SSL/TLS cert issued by a CA to be published in a public accessible log*

- *We can examine these logs with:*
  - https://censys.io
  - https://crt.sh

Information Gathering
Subdomain Enumeration - Passive

CT sources using curl:

```
$ curl -s "https://crt.sh/?q=${TARGET}&output=json" | jq -r '.[] | "\(.name_value)\n\(.common_name)"' | sort -u > "${TARGET}_crt.sh.txt“
```

curl -s	minimal output
https://crt.sh/?q=<DOMAIN>&output=json	Ask for the json output.
jq -r '.[]' "\(.name_value)\n\(.common_name)"'	Parse the output and print certificate's name value and common name one per line.
sort -u	sort and remove duplicates

Information Gathering
Subdomain Enumeration - Passive

CT sources using openssl:

```
$ openssl s_client -ign_eof 2>/dev/null <<<$'HEAD / HTTP/1.0\r\n\r' -connect "${TARGET}:${PORT}" | openssl x509 -noout -text -in - | grep 'DNS' | sed -e 's|DNS:|\n|g' -e 's|^\*.*||g' | tr -d ',' | sort -u
```

Information Gathering
Subdomain Enumeration - Passive

Automating Subdomain Enumeration

TheHarvester

- a modular effective tool for early-stage penetration testing and red team engagements

- collects emails, names, subdomains, IP addresses, and URLs from various public data sources for passive information gathering

Information Gathering
Subdomain Enumeration - Passive

TheHarvester - some modules:

Baidu	Baidu search engine.
Bufferoverun	Uses data from Rapid7's Project Sonar - www.rapid7.com/research/project-sonar/
Crtsh	Comodo Certificate search.
Hackertarget	Online vulnerability scanners and network intelligence to help organizations.
Otx	AlienVault Open Threat Exchange - https://otx.alienvault.com
Rapiddns	DNS query tool, which makes querying subdomains or sites using the same IP easy.
Virustotal	Domain search.

Information Gathering
Subdomain Enumeration - Passive

TheHarvester - some modules:

Threatcrowd	Open source threat intelligence.
Threatminer	Data mining for threat intelligence.
Trello	Search Trello boards (Uses Google search)
Urlscan	A sandbox for the web that is a URL and website scanner.
Vhost	Bing virtual hosts search.
Virustotal	Domain search.
Zoomeye	A Chinese version of Shodan.

Information Gathering
Subdomain Enumeration - Passive

TheHarvester - Usage example:

1. Create a file picking some of the sources names line by line

2. Run the tool with the command:

```
$ cat sources.txt | while read source; do theHarvester -d "${TARGET}" -b $source -f "${source}_${TARGET}";done
```

This will create result files in Json format

3. Extract all the subdomains found with:

```
$ cat *.json | jq -r '.hosts[]' 2>/dev/null | cut -d':' -f 1 | sort -u > "${TARGET}_theHarvester.txt"
```

Information Gathering
Ifrastructure enumeration - Passive

Netcraft - information about the servers without interacting with them

Background	General information (date the domain was first seen by Netcraft crawlers)
Network	netblock owner, hosting company, nameservers, etc.
Hosting history	Latest IPs used, webserver, and target OS.

latest IPs used: may expose the actual IP of the webserver before it was placed behind a load balancer, WAF, IDS…

Information Gathering
Ifrastructure enumeration - Passive

Internet Archive: a US digital library that provides free public access to digitalized materials, including websites, collected via its web crawlers.

With Wayback Machine we can find old versions that may have interesting comments in the source code or files that should not be there. This tool can be used to find older versions of a website at a point in time.

Searching for it using Wayback Machine and can find a version that utilizes a specific (now vulnerable) plugin.

Back to the current version of the site, we can examine if the plugin was not removed properly and can still be accessed (for example via the wp-content directory in wordpress).

We can also use waybackurls tool for urls of Wayback Machine

```
$ go install github.com/tomnomnom/waybackurls@latest
$ waybackurls -dates https://facebook.com > waybackurls.txt
$ cat waybackurls.txt
```

Information Gathering
Infrastructure enumeration - Active

Web Servers - discover as much information as possible from the webserver to understand its functionality (URL rewriting functionality,

Load balancing, script engines, IDS

Look at the response headers and inspect them manually :

```
$ curl -I "http://${TARGET}“
```

X-Powered-By header: tells us what the web app is using

Cookies: each technology by default has its cookies

…

Information Gathering
Infrastructure enumeration - Active

Probing web servers with tools that analyze their characteristics

- Whatweb:

  $ whatweb -a3 https://www.facebook.com –v

- Wappalyzer : a browser extension
- WafW00f: a web application firewall (WAF) fingerprinting tool that analyses responses to enumerate security solutions in place
  - *$ sudo apt install wafw00f –y*
  - *$ wafw00f -v https://www.target.com*
- Aquatone: automatic and visual inspection of websites across many hosts

Information Gathering
Subdomain Enumeration - Active

We can probe the infrastructure identified previously (organization and 3rd party DNS servers)

ZoneTransfers:

- how a secondary DNS server receives information from the primary DNS server and updates it.

- master-slave approach: master DNS should be configured to enable zone transfers from slave (secondary) DNS servers. This may be misconfigured

- https://hackertarget.com/zone-transfer/ Obtain info using the domain name we want to search
- Manually with nslookup:

$ nslookup -type=NS target.com – identify nameservers for target.com domain

$ nslookup -type=any -query=AXFR target.com <nameserver identified>

Information Gathering
Subdomain Enumeration - Active

vHosts : allow several websites to be hosted on a single server

- IP-based: a host with multiple network interfaces (or aliases)
- Name-based virtual hosting: multiple domain names configured at the application layer

vHost Fuzzing with vhost_list file:

```
$ cat ./vhosts | while read vhost_list;do echo "\n*********\nFUZZING: ${vhost}\n*********";curl -s -I http://<target IP>  -H "HOST: ${vhost}.sometarget.com" | grep "Content-Length: ";done
```

Automated vhost fuzzing with ffuf

```
$ ffuf -w ./vhosts -u http://<target IP> -H "HOST: FUZZ.sometarget.com" -fs 612
```

-w: Path to our wordlist

-u: URL we want to fuzz

-H "HOST: FUZZ.sometarget.com": HOST Header, FUZZ = the fuzzing point.

-fs 612: Filter responses with a size of 612.

Information Gathering
Subdomain Enumeration - Active

Crawlers

- Zed Attack Proxy – by OWASP: perform manual and automated security testing on web applications
- *ffuf – for crawling hidden folders (missed by ZAP)*

- *CeWL – extract keywords from a website to give them as input to ffuf for targeted fuzzing*

External Reconnaissance

Social Media

- LinkedIn, Facebook and Twitter can be a goldmine of information
  - Automated scraping tools such as LinkedInt may be used.
  - they often violate their user agreements (use accounts that may be “burned”)

- Websites such as hunter.io can be used to discover email addresses of employees.
  - Common patterns for employees may be found in order to guess valid emails.

Network Enumeration

Enumeration:

- the difficulty and the goal is that we must find the targets and identify all the ways we could attack them. Not to gain access to our target
- Tools are only useful if pentesters know what to do with the information they get from them. They cannot replace our knowledge
- Here (nmap) is much more about actively interacting with the services to see what information we could get and the possibilities they offer
- understand the technologies, protocols how they work and adapt to already acquired knowledge base
- Enumeration **is collecting as much information as possible**. The more information we have, the easier it will be to find vectors of attack.

Network Enumeration

Enumeration:

- Most of the info we get comes from misconfigurations due to ignorance or wrong security mindset
  - Example: an IT Admin only relies on a Firewall, GPOs and updating. Is this enough to secure the nerwork?
- Manual enumeration is critical: scanning tools can’t always bypass security measures.
  - What if a port is marked as closed (due to time laps of scanning response) and not as filtered or unknown? We may loose an opportunity to find a way to access the target.
- A time consuming (some times trial and error) procedure

Network Enumeration
Nmap (Network Mapper)

- an open-source network analysis and security auditing tool
- written in C, C++, Python, and Lua
- Scan networks to identify:
  - available hosts using raw packets
  - services and applications running (including versions)
  - Operating systems and versions
  - Can detect packet filtering (Firewalls, IDS) in place

Nmap – Use Cases

- Widely used by Admins or IT security specialists:
  - Audit the security aspects of networks
  - Simulate penetration tests
  - Check firewall and IDS settings and configurations
  - Types of possible connections
  - Network mapping
  - Response analysis
  - Identify open ports
  - Vulnerability assessment

Nmap – Use Cases

Offers many different types of scans that can be divided into the following techniques:

- Host discovery
- Port scanning
- Service enumeration and detection
- OS detection
- Nmap Scripting Engine – Interact with the target using scripts in the database

Nmap

Basic Syntax:

```
$ nmap <scan types> <options> <target>
```

Help:

```
$ nmap –help
```

Nmap

Different types of connections using differently structured packets to send:

-sS/sT/sA/sW/sM: TCP SYN/Connect()/ACK/Window/Maimon scans

-sU: UDP Scan

-sN/sF/sX: TCP Null, FIN, and Xmas scans

--scanflags <flags>: Customize TCP scan flags

-sI <zombie host[:probeport]>: Idle scan

-sY/sZ: SCTP INIT/COOKIE-ECHO scans

-sO: IP protocol scan

-b <FTP relay host>: FTP bounce scan

Nmap

-sS (default) :

- scan several thousand ports per second
- sends one packet with the SYN flag
  - never completes the three-way handshake
  - doesn’t establish a full TCP connection to the scanned port

- if a packet with the SYN-ACK flag is sent back by the target then the port will be flagged as open.
- If an RST flag is received back then it is assumed as closed.
- If no packet is received back it will be displayed as filtered. Packets may be dropped or ignored by a firewall

Nmap
Host Discovery

For an internal pentest on the entire network we should first actively discover all the systems we can interact with. Nmap has many methods for that. The most effective host discovery method is using ICMP echo requests.

Scan a network range example:

```
sudo nmap 10.10.1.0/24 -sn -oA live_hosts | grep for | cut -d" " -f5
```

10.10.1.0/24: network range to scan for live systems

-sn: without port scanning

-oA live_hosts: save the results in all formats with the name live_hosts

***This works if the firewalls are not configured to block our traffic. In this case some other techniques may be used to see if the systems are up on not.*

Nmap
Host Discovery

Scan a list of IPs: Very common scenario where we are given specific IPs

```
sudo nmap -sn -oA scan_results -iL hosts_lists.txt | grep for | cut -d" " -f5
```

-iL: scan a given list of target IPs (line by line format)

Scan multipe IPs:

```
sudo nmap -sn -oA multi-IP_scan 10.10.1.1 10.20.1.1 10.30.1.1-20 | grep for | cut -d" " -f5
```

10.10.1.1...: network hosts to scan

-sn: without port scanning - ICMP Echo Requests (-PE)

-oA host: save the results in all formats with the name live_hosts

An ICMP reply is expected back if the host is alive. In any other case Nmap marks hosts as inactive. But in order to do that we have to define (--PE) option.

-PE: ping scan using 'ICMP Echo requests'

-packet-trace: show all packets sent and received

--reason: describe which discovery test(s) the host responded to. normal Nmap output only indicates whether a host is up or not

--disable-arp-ping: disable ARP requests and scan with ICMP echo requests

Nmap
Port Scanning

If we know how the tool works then we can understand the results.

Till now we found the systems that are alive. Now we want to get more details.

We need:

- Open ports and its services
- Service versions and information they provide
- Operating systems

…

Nmap
Port States

State	Description
open	the connection to the port has been established. Can be **TCP connections**, **UDP datagrams** as well as **SCTP associations**.
closed	the TCP protocol indicates that the received packet contains an RST flag. This can also be used to determine if our target is alive or not.
filtered	Nmap cannot identify whether the port is open or closed Either no response is returned back or we get an error code from the target.
unfiltered	Usually occurs during the **TCP-ACK** scan indicates that the port is accessible, but it is not clear whether it is open or closed.
open|filtered	Nmap will set that If a response is not received. This indicates that a firewall or packet filter may protect the port.
closed|filtered	This state only occurs in the **IP ID idle** scans and indicates that it was impossible to determine if the scanned port is closed or filtered (firewall block).

Nmap
Port Scanning TCP

- By default Nmap scans top 1000 TCP ports with SYN scan (-sS)
- SYN Scan requires root privs because it needs socket permissions to craft raw TCP packets
- We can define ports
  - one by one: -p 22,25,80….
  - by range: -p 22-445
  - top n ports: --top-ports==n (n = whaterver)
  - Fast port scanning: -F
  - All ports: -p-

Example:

```
sudo nmap 10.10.1.28 -p 21 --packet-trace -Pn -n --disable-arp-ping
```

Here to have a clear view of the scan we disable ICMP echo requests (-Pn), DNS resolution (-n) and ARP ping scan (--disable-arp-ping).

sudo nmap 10.129.2.28 -p 443 --packet-trace --disable-arp-ping -Pn -n --reason -sT

Nmap
Port Scanning TCP – Connect Scan

TCP Connect Scan (-sT)

- uses the TCP three-way handshake to determine if a specific port on a target host is open or closed. Considered open if the target responds with an SYN-ACK packet and closed if it responds with an RST packet

- it is the most accurate way to determine the state of a port
- most stealthy: unlike other types of scans (SYN) it doesn't leave any unfinished connections, or unsent packets → less likely to be detected by IDS/IPS
- Slower that other types because it requires to wait for a response for each packet it sends

Example:

sudo nmap 10.10.1.21 -p 443 --packet-trace --disable-arp-ping -Pn -n --reason -sT

Nmap
Open UDP Ports

- System Admins can forget to filter UDP ports.
- UDP is stateless (no three-way handshaking) – no ACK responses, so the timeout is much longer and the UDP scan much slower.

  Example:

  sudo nmap 10.10.1.1 -F –sU

  -F : scan top 100 ports

  -sU: UDP Scan

- Nmap packets are sent as empty datagrams – if the Port is open it must also be configured to respond
- an ICMP response with error code 3 (port unreachable) shows that the port is closed.

  For all other ICMP responses, the scanned ports are marked as (open|filtered).

Nmap
Version Scan

- A very useful option for scanning ports is -sV

```
sudo nmap 10.10.1.1 -Pn -n --disable-arp-ping --packet-trace -p 445 --reason  -sV
```

-sV: option is used to get additional information from open ports like versions, service names, and details about our target.

Nmap Scripting Engine

With NSE we can create scripts in Lua to interact with certain services. These scripts can be divided into 14 categories:

Category	Description
auth	Determination of authentication credentials.
broadcast	Scripts, which are used for host discovery by broadcasting and the discovered hosts, can be automatically added to the remaining scans.
brute	Executes scripts that try to log in to the respective service by brute-forcing with credentials.
default	Default scripts executed by using the -sC option.
discovery	Evaluation of accessible services.
dos	These scripts are used to check services for denial of service vulnerabilities and are used less as it harms the services.
exploit	This category of scripts tries to exploit known vulnerabilities for the scanned port.

Nmap Scripting Engine

With NSE we can create scripts in Lua to interact with certain services. These scripts can be divided into 14 categories:

Category	Description
external	Scripts that use external services for further processing.
fuzzer	This uses scripts to identify vulnerabilities and unexpected packet handling by sending different fields, which can take much time.
intrusive	Intrusive scripts that could negatively affect the target system.
malware	Checks if some malware infects the target system.
safe	Defensive scripts that do not perform intrusive and destructive access.
version	Extension for service detection.
vuln	Identification of specific vulnerabilities.

Nmap
Scripting Engine

Examples:

Default scripts: *$sudo nmap <target> -sC*

Specific scripts category: *$ sudo nmap <target> --script <category>*

Specific script: *$ sudo nmap 10.129.2.28 -p 25 --script banner,smtp-commands*

Aggressive scan: $ sudo nmap 10.129.2.28 -p 80 –A (extra information about webservers, web apps running, os info)

Vulnerability Assessment: Further scan with scripts to verfiy vulnerabilities (for example on database servers used by a web application)

$ sudo nmap 10.129.2.28 -p 80 -sV --script vuln

Nmap Results

Nmap can save the results in 3 different formats.

- Normal output (-oN) - .nmap file extension
- Grepable output (-oG) - .gnmap file extension
- XML output (-oX) - .xml file extension

-oA option: save the results to all formats.

Nmap Performance

- **Timeouts** – min and max RTT (Round-Trip-Times) default is 100ms

  Decreasing the max RTT value to optimize our scan may end up overlooking some hosts

- **Max Retries**

  Default retry rate is 10. After that, nmap does not send any more packets and skips the port.

  Reducing retries may also have negative effect on our results, overlooking targets

- **Rates**

- Default timing template (default is -T 3)
  - -T 0 / -T paranoid
  - -T 1 / -T sneaky
  - -T 2 / -T polite
  - -T 3 / -T normal
  - -T 4 / -T aggressive
  - -T 5 / -T insane

  Values 0-5 : Aggressiveness of scanning. May produce heavy network traffic and security systems may block the scan

Nmap
IDS/IPS and Firewall

Firewall:

- a security mechanism against unauthorized connections from external networks that could be potentially dangerous
- a software based solution that monitors incoming network traffic and decides how to handle the connections based on the rules

IDS/IPS (Intrusion Detection/Prevention system)

- IDS – scans the network to identify potential attacks, analyzes them and reports them
- IPS – takes extra defensive measures when an attack is detected
- Pattern matching and signature based analysis
- If a pattern is matched (like a service detection scan) IPS may prevent pending connection attempts

Nmap
IDS/IPS and Firewall Evasion

Firewalls and rules detection

- Dropped packets are ignored, no response is returned
- Rejected packets are returned with an RST flag containing ICMP error codes - or nothing
  - Net Unreachable
  - Net Prohibited
  - Host Unreachable
  - Host Prohibited
  - Port Unreachable
  - Proto Unreachable
- With Nmap:
  - TCP-ACK scan (-sA): much harder for firewalls and IDS/IPS to filter. Only a packet with the ACK flag is sent.
  - SYN (-sS) or Connect Scans (-sT): easyly filtered. If the port is closed or open the host will respond with an RST flag.
  - All incoming connections with the SYN flag are usually blocked by firewalls
  - Packets with ACK flag are often passed by the firewall because it cannot determine what is the source of the connection establishment (internal or external network)

- We can try these in a firewall filtered network to observe differences on the RCVD packets and the flags set (R, SA):
  - SYN-Scan: *$ sudo nmap <IP> -p <Ports> -sS -Pn -n --disable-arp-ping --packet-trace*
  - ACK-Scan: *$ sudo nmap <IP> -p <Ports> -sA -Pn -n --disable-arp-ping --packet-trace*

.

Nmap
IDS/IPS and Firewall Evasion

IDS/IPS Detection

- Detection is much more difficult (than firewalls) because they monitor traffic **passively**
- **IDS** system: examine all connections between hosts. If it finds packets containing specific content, it sends notifications(alerts) usually to the admins group to take actions. We can try to trigger these actions by (for example) aggressively scanning a single port and service. Based on the measures taken we can detect if there is a monitoring application in place or not.
- **IPS** system: automatically takes measures configured by the admin to prevent potential attacks. An additional application to the IDS. We can detect them by scanning from a specific single host (VPS) and check if it will get blocked at any time.
- The use of **VPS** ranges to detect these measures on a target network during an assessment is mandatory. In case we are detected, our IP will be blocked by the admins so we won’t be able to access the network using the blocked IPs. Even worst the ISP will be eventually contacted and our IP will be blacklisted globally (…Abuse IP DB check).
- Our goal is to go under the radar with our scans (be quieter) in order to be able to continue with our pentest

Nmap
IDS/IPS and Firewall Evasion

Decoy Scanning

- What if the IPS prevents any access on the network from specific subnets (from different regions)?
- Decoy scanning with nmap:
  - Randomly generate a number of IPs and use them in spoofed packets for scanning
  - Our real IP will be randomly placed somewhere between these IPs
  - Decoys must be alive in order to be able to interact with the service if needed. Otherwise we may be recognized and blocked as SYN-flooding attack.

  Ex:
  ```
  $ sudo nmap <IP> -p <Ports> -sS -Pn -n --disable-arp-ping --packet-trace -D RND:5
  ```

  - We can also specify a source IP address with the –S flag if we have an indication that specific subnets may be allowed.

  Ex:
  ```
  $ sudo nmap <IP> -n -Pn -p <port> -O -S <spoofed IP address> -e tun0
  ```

Nmap
IDS/IPS and Firewall Evasion

DNS Proxying

- By default nmap performs DNS resolution
- DNS queries are by default whitelisted
- DNS queries go over UDP port 53
- TCP port 53 : Used for "Zone Transfers" or large data transfers
- Due to IPv6 and DNSSEC expansion many DNS requests go via TCP port 53

- Nmap allows to specify the DNS servers : --dns-server <ns> . This could be extremely useful for our scans if we are placed in DMZ and use the company's trusted DNS servers and use them to try to interact with hosts on the internal network
- We could also specify TCP Port 53 as a source port (--source-port) for our scans in case it is not properly filtered by IDS/IPS measures.

Ex:

```
$ sudo nmap <IP> -p<Filtered Port> -sS -Pn -n --disable-arp-ping --packet-trace --source-port 53
```

Nmap
IDS/IPS and Firewall Evasion

HTB Platform

Questions?