

# S9-L5

## Traccia: Threat Intelligence & IOC

Durante la lezione teorica, abbiamo esaminato il concetto di **Threat Intelligence** e gli **Indicatori di Compromissione (IOC)**. Abbiamo appreso che gli **IOC** sono evidenze o eventi che indicano un attacco in corso o già avvenuto.

## Esercizio: Threat Intelligence & IOC

Per l'esercizio pratico di oggi, è allegata una **cattura di rete** effettuata con **Wireshark**. Analizzate attentamente la cattura e rispondete ai seguenti quesiti:

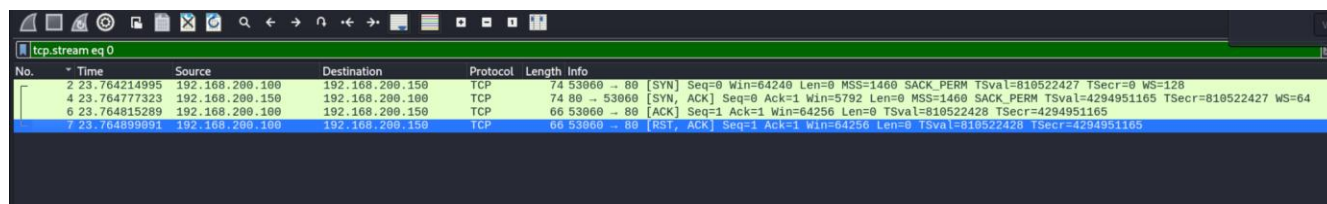
1. **Identificare ed analizzare eventuali IOC**, ovvero evidenze di attacchi in corso.
2. **Formulare ipotesi sui potenziali vettori di attacco** utilizzati in base agli IOC trovati.
3. **Suggerire azioni per ridurre l'impatto dell'attacco attuale** e prevenire attacchi simili in futuro.

## Svolgimento

Inizio scaricando il file fornito dalla consegna e copiandolo all'interno della cartella condivisa preventivamente creata.

## Analisi del traffico con Wireshark

Apro il file con wireshark e inizio controllando le statistiche sui protocolli di rete andando in **Wireshark > Statistics > Protocol Hierarchy**. In questo modo posso vedere quali sono i protocolli più utilizzati nel frammento di traffico catturato in questo file.



No.	Time	Source	Destination	Protocol	Length	Info
2	23.764214995	192.168.200.100	192.168.200.150	TCP	74	53060 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=810522427 TSecr=0 WS=128
4	23.764777323	192.168.200.150	192.168.200.100	TCP	74	80 → 53060 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM TSval=4294951165 TSecr=810522427 WS=64
6	23.764815289	192.168.200.100	192.168.200.150	TCP	66	53060 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=810522428 TSecr=4294951165
7	23.764899991	192.168.200.100	192.168.200.150	TCP	66	53060 → 80 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=810522428 TSecr=4294951165

Wireshark - Protocol Hierarchy Statistics - progetto.pcapng

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s	PDUs
Frame	100.0	2083	100.0	139872	30 k	0	0	0	2083
Ethernet	100.0	2083	25.2	35276	7,652	0	0	0	2083
Internet Protocol Version 4	99.8	2079	29.7	41580	9,019	0	0	0	2079
User Datagram Protocol	0.0	1	0.0	8	1	0	0	0	1
NetBIOS Datagram Service	0.0	1	0.2	244	52	0	0	0	1
SMB (Server Message Block Protocol)	0.0	1	0.1	162	35	0	0	0	1
SMB MailSlot Protocol	0.0	1	0.0	25	5	0	0	0	1
Microsoft Windows Browser Protocol	0.0	1	0.1	76	16	1	76	16	1
Transmission Control Protocol	99.8	2078	44.8	62652	13 k	2078	62652	13 k	2078
Address Resolution Protocol	0.2	4	0.1	148	32	4	148	32	4

No display filter.

Close Copy Protocols Help

La maggior parte, oltre il 99%, avviene su TCP. Controllo anche gli indirizzi IP coinvolti.

Controllo controllando sempre nelle statistiche, ma relative agli endpoint. Li trovo in **Statistics > Endpoints**

Ethernet · 3		IPv4 · 3	IPv6	TCP · 2015	UDP · 2	
Address	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes
08:00:27:39:7d:fe	2,082	140 kB	1,054	78 kB	1,028	62 kB
08:00:27:fd:87:1e	2,083	140 kB	1,029	62 kB	1,054	78 kB
ff:ff:ff:ff:ff:ff	1	286 bytes	0	0 bytes	1	286 bytes

Questi endpoint corrispondono a due indirizzi IP: **192.168.200.100** e **192.168.200.150**.

Wireshark · Endpoints · progetto.pcapng

Endpoint Settings

☐ Name resolution

☐ Limit to display filter

Copy

Map

Protocol

☐ Bluetooth

☐ BPv7

☐ DCCP

☒ Ethernet

☐ FC

☐ FCoE

Filter list for specific type

Ethernet · 3IPv4 · 3IPv6TCP · 2015UDP · 2

Address	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes
192.168.200.100	2,078	139 kB	1,052	78 kB	1,026	62 kB
192.168.200.150	2,079	140 kB	1,027	62 kB	1,052	78 kB
192.168.200.255	1	286 bytes	0	0 bytes	1	286 bytes

× Close

Help

La macchina attaccante è **192.168.200.100**, mentre la vittima è **192.168.200.150**.

Analizzando il pacchetto numero 1 è possibile identificare la macchina vittima come macchina metasploitable

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	SPort	DPort	Info
1	0.000000000	192.168.200.100	192.168.200.150	BROWSER	266			Host Announcement METASPLOITABLE, Workstation, Server, Print Queue Server, Xenix Ser
2	23.764214995	192.168.200.100	192.168.200.150	TCP	74	53060	80	53060 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=819522427 TSecr=0 W
3	23.764297788	192.168.200.150	192.168.200.100	TCP	74	33876	443	33876 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=819522428 TSecr=0 W
4	23.764777323	192.168.200.150	192.168.200.100	TCP	74	80	53060	80 → 53060 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM TSval=4294951165
5	23.764777427	192.168.200.150	192.168.200.100	TCP	60	443	33876	443 → 33876 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
6	23.764815288	192.168.200.100	192.168.200.150	TCP	66	53060	80	53060 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=819522428 TSecr=4294951165
7	23.764899891	192.168.200.100	192.168.200.150	TCP	66	53060	80	53060 → 80 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=819522428 TSecr=4294951165

Analizzando il traffico si può notare che si tratta di un three-way handshake del protocollo TCP.

Il three-way handshake è il processo con cui il protocollo TCP stabilisce una connessione affidabile tra due dispositivi prima di trasmettere i dati. Questo garantisce che entrambi gli endpoints siano pronti a comunicare.

Le tre fasi sono le seguenti:

1. **SYN (Synchronize)** → Il client (attaccante) invia un pacchetto **SYN** al server per richiedere la connessione.
2. **SYN-ACK (Synchronize-Acknowledge)** → Il server (vittima) risponde con un pacchetto **SYN-ACK** per accettare la richiesta.
3. **ACK (Acknowledge)** → Il client (attaccante) invia un pacchetto **ACK** per confermare la connessione, completando il processo.

Nel caso in cui la connessione venga conclusa con l'handshake, viene inviato un pacchetto RST, ACK che interrompe la connessione appena conclusa post handshake.

```

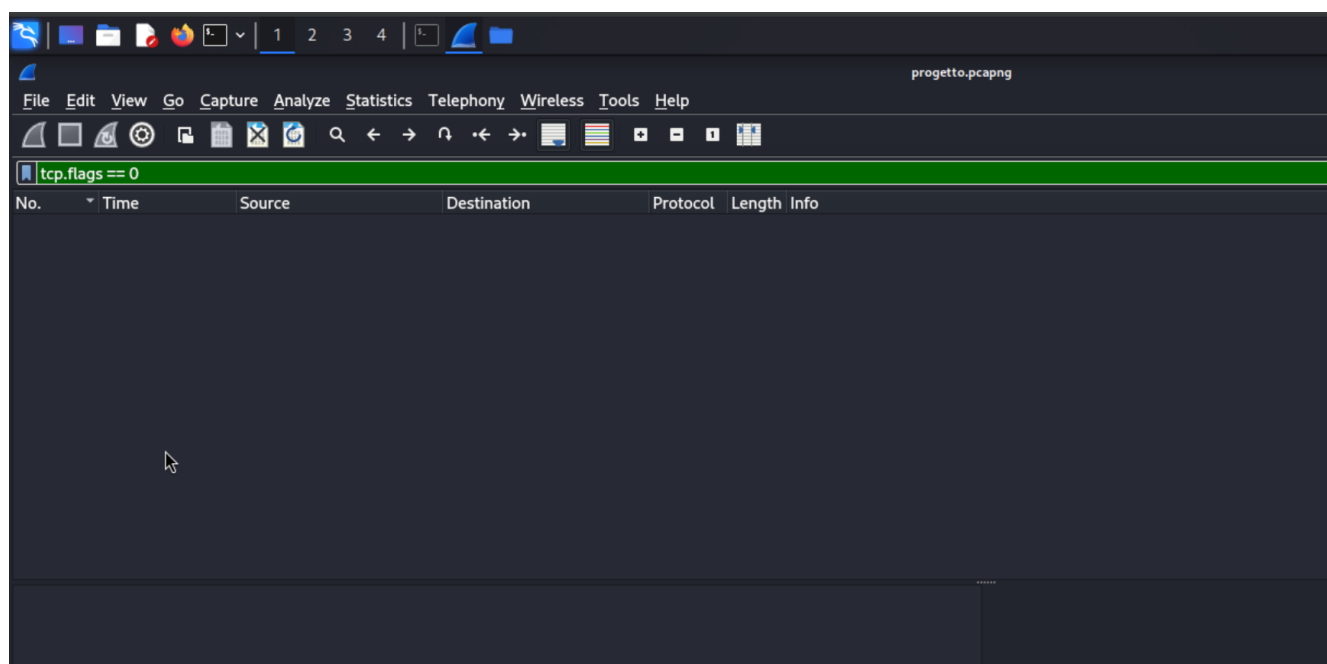
2 → 53050 [SYN, ACK] Seq=
0 → 53062 [SYN, ACK] Seq=
5656 → 22 [ACK] Seq=1 Ack
3062 → 80 [ACK] Seq=1 Ack
1182 → 21 [RST, ACK] Seq=
5656 → 22 [RST, ACK] Seq=
3062 → 80 [RST, ACK] Seq=
0684 → 199 [SYN] Seq=0 Wi

```

Per analizzare più comodamente i pacchetti uso i filtri.

Inizio controllando se ci sono pacchetti che utilizzano pacchetto TCP senza flag normali

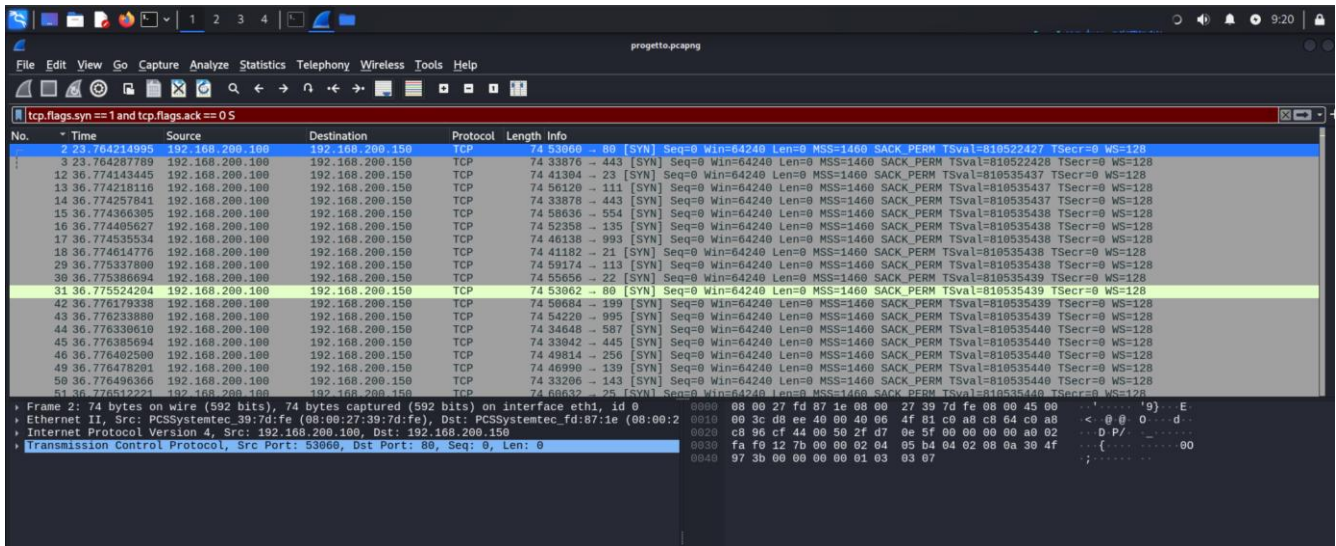
`tcp.flags == 0`



Non ce ne sono.

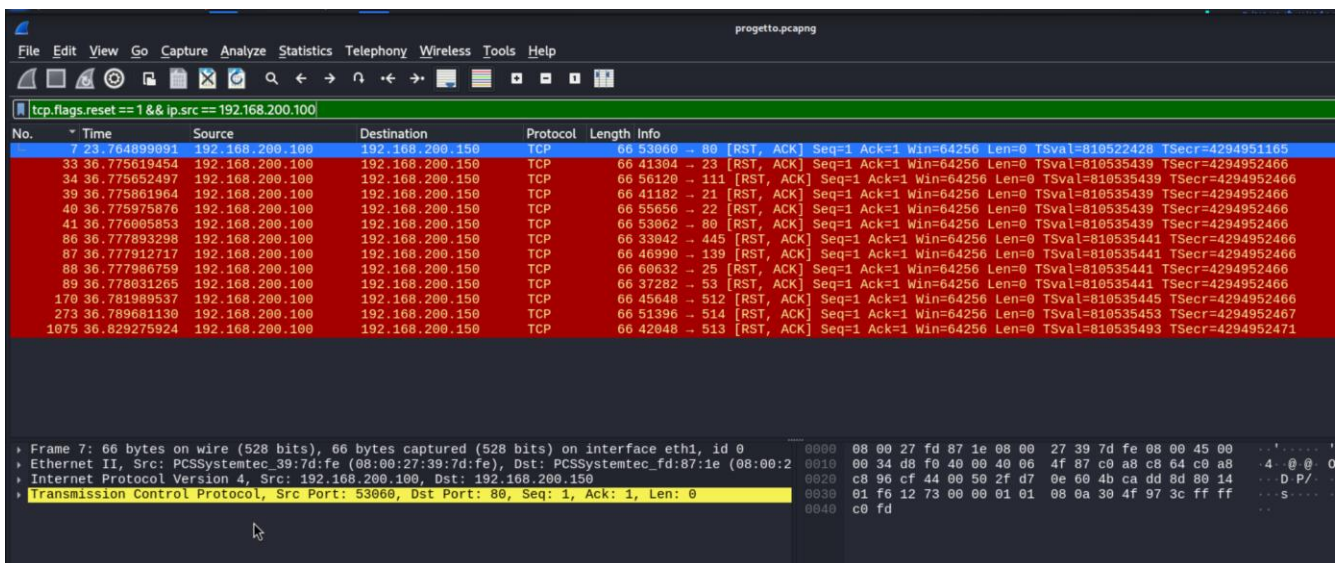
Controllo ora i pacchetti TCP di SYN senza ACK.

`tcp.flags.syn == 1 and tcp.flags.ack == 0`



Ce ne sono tantissimi. Questo comportamento è tipico di una scansione per individuare **servizi aperti** su un target.

Controllo ora i pacchetti che hanno ricevuto un RST, cioè quelli a cui è stata interrotta la connessione



Su queste porte sono state chiuse le connessioni. Sono le porte **80, 23, 111, 21, 22, 445, 139, 512, 513, 514**.

È utile anche seguire lo stream TCP di un pacchetto cliccando il tasto destro su un pacchetto, per vedere lo storico della singola connessione stabilita e resettata.

Potrebbe trattarsi di un nmap -sS (stealth) o di un RST flood

Per escludere il secondo uso questo comando

`tcp.flags.reset == 1 && !tcp.flags.syn`

`tcp.flags.reset == 1` → Mostra solo i pacchetti con **flag RST attivo**.

!tcp.flags.syn → Esclude tutti i pacchetti che contengono il **flag SYN**.

Analizzo alcuni degli stream (connessioni effettuate con successo):

tcp.stream eq 0						
No.	Time	Source	Destination	Protocol	Length	Info
2	23.764214995	192.168.200.100	192.168.200.150	TCP	74	53060 → 80 [SYN] S
4	23.764777323	192.168.200.150	192.168.200.100	TCP	74	80 → 53060 [SYN, A
6	23.764815289	192.168.200.100	192.168.200.150	TCP	66	53060 → 80 [ACK] S
7	23.764899091	192.168.200.100	192.168.200.150	TCP	66	53060 → 80 [RST, A

tcp.stream eq 1						
No.	Time	Source	Destination	Protocol	Length	Info
3	23.764287789	192.168.200.100	192.168.200.150	TCP	74	33876 → 443 [SYN] S
5	23.764777427	192.168.200.150	192.168.200.100	TCP	60	443 → 33876 [RST, A

tcp.stream eq 2						
No.	Time	Source	Destination	Protocol	Length	Info
12	36.774143445	192.168.200.100	192.168.200.150	TCP	74	41304 → 23 [SYN] S
19	36.774685505	192.168.200.150	192.168.200.100	TCP	74	23 → 41304 [SYN, A
24	36.774700464	192.168.200.100	192.168.200.150	TCP	66	41304 → 23 [ACK] S
33	36.775619454	192.168.200.100	192.168.200.150	TCP	66	41304 → 23 [RST, A

Si ipotizza quindi un **nmap stealth** da parte della macchina attaccante (ip 192.168.200.100) versi la macchina vittima (IP 192.168.200.150, identificata come metasploitable) con servizi attivi sulle seguenti porte: 80, 23, 111, 21, 22, 445, 139, 512, 513, 514