

THREAT INTELLIGENCE & IOC

S9/L3

TABLE OF CONTENT

03. INTRODUCTION

04. PROCEDURE

INTRODUCTION

Traccia:

Durante la lezione teorica, abbiamo visto la Threat Intelligence e gli indicatori di compromissione.

Abbiamo visto che gli IOC sono evidenze o eventi di un attacco in corso, oppure già avvenuto.

Per l'esercizio pratico di oggi, trovate in allegato una cattura di rete effettuata con Wireshark.

Analizzate la cattura attentamente e rispondere ai seguenti quesiti:

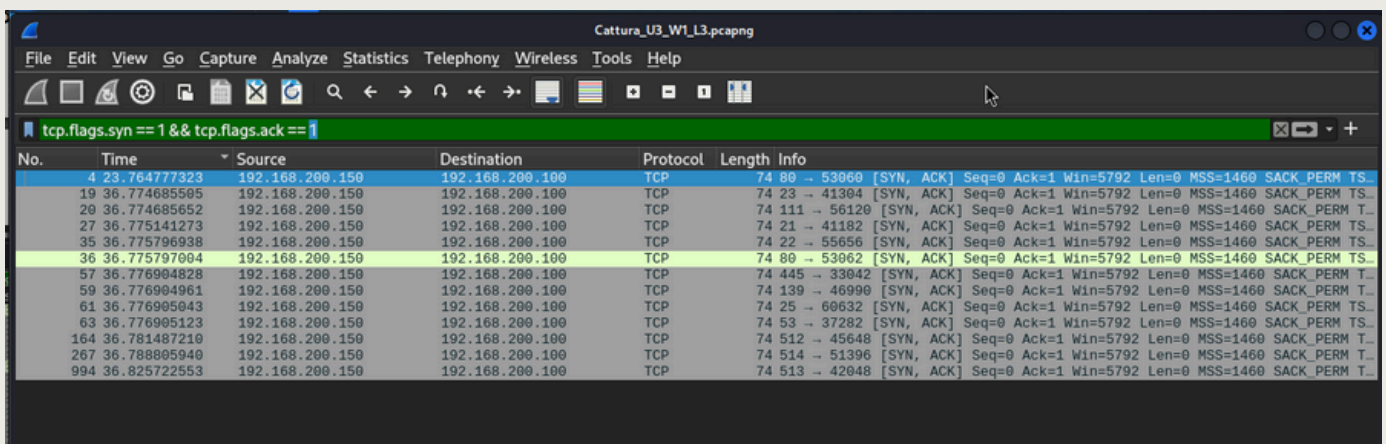
- Identificare eventuali IOC, ovvero evidenze di attacchi
- In corso In base agli IOC trovati, fate delle ipotesi sui potenziali vettori di attacco utilizzati
- Consigliate un'azione per ridurre gli impatti dell'attacco

PROCEDURE

Per facilitare la navigazione all'interno di questo file, possiamo utilizzare dei filtri.

Il primo filtro ci aiuterà a determinare il numero di pacchetti (o frame) catturati dalla rete che sono arrivati alla seconda fase del three-way handshake (SYN-ACK).

Il comando per applicare questo filtro è:
tcp.flags.syn == 1 && tcp.flags.ack == 1



The image shows a Wireshark packet capture window titled "Cattura_U3_W1_L3.pcapng". The filter bar at the top contains the filter expression `tcp.flags.syn == 1 && tcp.flags.ack == 1`. Below the filter, a list of captured packets is displayed. The table has columns for No., Time, Source, Destination, Protocol, Length, and Info. The packets are all TCP SYN-ACKs from 192.168.200.150 to 192.168.200.100. Packet 36 is highlighted in yellow.

No.	Time	Source	Destination	Protocol	Length	Info
4	23.764777323	192.168.200.150	192.168.200.100	TCP	74	80 → 53960 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM TS...
19	36.774685505	192.168.200.150	192.168.200.100	TCP	74	23 → 41304 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM TS...
20	36.774685652	192.168.200.150	192.168.200.100	TCP	74	111 → 56120 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM T...
27	36.775141273	192.168.200.150	192.168.200.100	TCP	74	21 → 41182 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM TS...
35	36.775796938	192.168.200.150	192.168.200.100	TCP	74	22 → 55656 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM TS...
36	36.775797004	192.168.200.150	192.168.200.100	TCP	74	80 → 53962 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM TS...
57	36.776904828	192.168.200.150	192.168.200.100	TCP	74	445 → 33042 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM T...
59	36.776904961	192.168.200.150	192.168.200.100	TCP	74	139 → 46990 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM T...
61	36.776905043	192.168.200.150	192.168.200.100	TCP	74	25 → 60632 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM TS...
63	36.776905123	192.168.200.150	192.168.200.100	TCP	74	53 → 37282 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM TS...
164	36.781487210	192.168.200.150	192.168.200.100	TCP	74	512 → 45648 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM T...
267	36.788805940	192.168.200.150	192.168.200.100	TCP	74	514 → 51396 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM T...
994	36.825722553	192.168.200.150	192.168.200.100	TCP	74	513 → 42048 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM T...

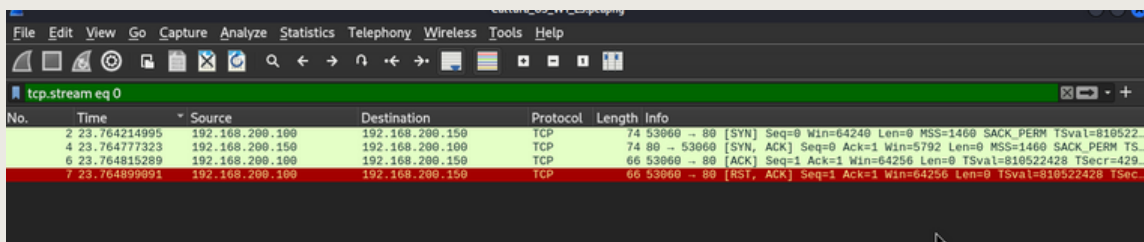
PROCEDURE

Se clicchiamo con il tasto destro su un frame e selezioniamo "Follow TCP", vedremo l'output dell'immagine sottostante.

Tutte le porte nella prima immagine produrranno un output strutturato in quattro frame come quello mostrato.

Possiamo osservare:

- **Primo frame:** Questo è il primo pacchetto inviato dal client al server per avviare una connessione TCP. Nel pacchetto SYN, il client invia un numero di sequenza iniziale (ISN) e richiede l'avvio di una connessione.
- **Secondo frame:** Questo è il secondo pacchetto, inviato dal server in risposta al pacchetto SYN del client. Il server risponde con un proprio numero di sequenza e un numero di acknowledgment (ACK) che conferma la ricezione del SYN da parte del client.
- **Terzo frame:** Questo è il terzo pacchetto, inviato dal client per confermare la ricezione del SYN-ACK del server. Con questo pacchetto, il client invia un acknowledgment per il numero di sequenza del server, completando così l'handshake.
- **Pacchetto in rosso:** Lo screenshot di Wireshark mostra un pacchetto TCP con i flag "RST" e "ACK" attivi. Questo tipo di pacchetto viene solitamente utilizzato per terminare una connessione TCP o per indicare che un pacchetto è stato ricevuto in un contesto inaspettato. Il flag RST (Reset) viene usato per interrompere una connessione, ad esempio, se un host riceve un pacchetto per una connessione che non riconosce o non desidera mantenere attiva. Il flag ACK (Acknowledge) conferma che il pacchetto è stato ricevuto.



The screenshot shows a Wireshark packet capture window with the title 'tcp.stream eq 0'. The packet list pane displays four packets. The first three packets (2, 4, and 6) are part of a TCP handshake: a SYN packet from 192.168.200.100 to 192.168.200.150, a SYN-ACK packet from 192.168.200.150 to 192.168.200.100, and an ACK packet from 192.168.200.100 to 192.168.200.150. The fourth packet (7) is a red packet, indicating a reset, with the flag 'RST, ACK' and sequence number 1. The packet details pane shows the 'Info' column for each packet, providing details such as sequence numbers, window sizes, and flags.

No.	Time	Source	Destination	Protocol	Length	Info
2	23.764214995	192.168.200.100	192.168.200.150	TCP	74	53960 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=810522...
4	23.764777323	192.168.200.150	192.168.200.100	TCP	74	80 → 53960 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM TS...
6	23.764815289	192.168.200.100	192.168.200.150	TCP	66	53960 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=810522428 TSecr=429...
7	23.764899991	192.168.200.100	192.168.200.150	TCP	66	53960 → 80 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=810522428 TSecr=...