## **S9 L3**

## Monitoraggio eventi e azioni preventive



**Daniele Zizzi** 

L'immagine mostra un file di cattura Wireshark che contiene un attacco DOS di tipo TCP Reset. L'attacco è stato generato dall'indirizzo IP 192.168.200.100 (attaccante), che appartiene alla stessa sottorete della macchina vittima, ovvero 192.168.200.150.

L'attaccante sta inviando pacchetti TCP con un window size di 64240 byte, che è molto più grande del MSS(Maximum Segment Size) di 1460 byte negoziato durante la fase di three-way handshake. Questo significa che l'attaccante sta inviando pacchetti che sono troppo grandi per essere gestiti dalla macchina vittima.

La macchina vittima, un server Metasploit, sta tentando di gestire i pacchetti in arrivo, ma non è in grado di farlo. Per questo motivo, sta resettando continuamente la connessione (RST, ACK).

Gli attacchi sono rivolti alle seguenti porte:

- 80 (HTTP)
- 443 (HTTPS)
- 21 (FTP)
- 22 (SSH)
- 23 (Telnet)

Questi attacchi possono avere un impatto significativo sulla macchina vittima, rendendola inutilizzabile.

Misure di protezione

Ecco alcune misure di protezione che possono essere adottate per mitigare gli attacchi TCP Reset:

- Utilizzare un firewall per bloccare i pacchetti TCP provenienti da indirizzi IP sconosciuti.
- Abilitare la funzione di protezione da attacchi TCP Reset sul server o sul dispositivo di rete.
- Aggiornare il software del server o del dispositivo di rete con le ultime patch di sicurezza.
- Load Balancing, in modo da distribuire l'attacco su più host e quindi ridurne l'entità.

Inoltre, è importante essere consapevoli di questo tipo di attacco e adottare misure di mitigazione preventive.

Poiché, in questo caso, l'attacco proviene da una sola macchina, possiamo bloccare l'IP usando il firewall, in modo da bloccare tutte le comunicazioni dall'attaccante. È buona norma utilizzare un SIEM o un SOAR. Nel primo caso verremmo solo avvertiti dell'attacco, mentre nel secondo il SOAR interverrebbe in automatico.

```
Interface name: eth1
Encapsulation type: Ethernet (1)
Arrival Time: Aug 9, 2022 11:59:23.658032486 CEST
[Time shift for this packet: 0.000000000 seconds]
Epoch Time: 1660039163.658032486 seconds
[Time delta from previous captured frame: 23.764214995 seconds]
[Time delta from previous displayed frame: 23.764214995 seconds]
[Time since reference or first frame: 23.764214995 seconds]
[Time since reference or first frame: 23.764214995 seconds]
[Frame length: 74 bytes (592 bits)
Capture Length: 74 bytes (592 bits)
[Frame is ignored: False]
[Frame is ignored: False]
[Protocols in frame: eth:ethertype:ip:tcp]
[Coloring Rule Name: HTTP]
[Coloring Rule String: http || tcp.port == 80 || http2]

Ethernet II, Src: PcsCompu_39:7d:fe (08:00:27:39:7d:fe), Dst: PcsCompu_fd:87:1e (08:00:27:fd:87:1e)

**Internet Protocol Version: 4

... 0101 = Header Length: 20 bytes (5)

**Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
Total Length: 60
Identification: 0xd8ee (55534)

**Olion: — Flags: 0x2, Don't fragment
... 0000 0000 0000 = Fragment Offset: 0
Time to Live: 64

Protocol: TCP (6)
Header Checksum: 0x4f81 [validation disabled]
[Header checksum: 0x4f81 [validation disabled]
[Header checksum: 10x161 [validation disabled]
Source Address: 192.168.200.160

Destination Address: 192.168.200.160

Destination Address: 192.168.200.150

**Transmission Control Protocol, Src Port: 53060, Dst Port: 86, Seq: 0, Len: 0
                                                                                                                                                                                                                                                                                                                                                     Wireshark - Packet 2 - Cattura_U3_W1_L3.pcapng
       Transmission Control Protocol, Src Port: 53060, Dst Port: 80, Seq: 0, Len: 0
Source Port: 53060
Destination Port: 80
             Destination Port: 80
[Stream index: 0]
[Conversation completeness: Complete, NO_DATA (39)]
[TCP Segment Len: 0]
Sequence Number: 0 (relative sequence number)
Sequence Number (raw): 802623071
[Next Sequence Number: 1 (relative sequence number)]
Acknowledgment Number: 0
Acknowledgment number (raw): 0
1010 .... = Header Length: 40 bytes (10)
Flags: 0x002 (SYN)
Window: 64240
                  Window: 64240
[Calculated window size: 64240]
Checksum: 0x127b [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0

√ Show packet bytes

                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 Close Help
```

21 36.774685696 192.168.200.150	192.168.200.100	TCP	60 443 → 33878 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
22 36.774685737 192.168.200.150	192.168.200.100	TCP	60 554 → 58636 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
23 36.774685776 192.168.200.150	192.168.200.100	TCP	60 135 → 52358 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
24 36.774700464 192.168.200.100	192.168.200.150	TCP	66 41304 → 23 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=810535438 TSecr=4294952466
25 36.774711072 192.168.200.100	192.168.200.150	TCP	66 56120 → 111 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=810535438 TSecr=4294952466
26 36.775141104 192.168.200.150	192.168.200.100	TCP	60 993 → 46138 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
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 TSval=4294952466 TSecr=810535438 WS=64
28 36.775174048 192.168.200.100	192.168.200.150	TCP	66 41182 → 21 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=810535438 TSecr=4294952466
29 36.775337800 192.168.200.100	192.168.200.150	TCP	74 59174 → 113 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=810535438 TSecr=0 WS=128
30 36.775386694 192.168.200.100	192.168.200.150	TCP	74 55656 → 22 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=810535439 TSecr=0 WS=128
31 36.775524204 192.168.200.100	192.168.200.150	TCP	74 53062 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=810535439 TSecr=0 WS=128
32 36.775589806 192.168.200.150	192.168.200.100	TCP	60 113 → 59174 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
33 36.775619454 192.168.200.100	192.168.200.150	TCP	66 41304 → 23 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=810535439 TSecr=4294952466
34 36.775652497 192.168.200.100	192.168.200.150	TCP	66 56120 → 111 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=810535439 TSecr=4294952466
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 TSval=4294952466 TSecr=810535439 WS=64
36 36.775797004 192.168.200.150	192.168.200.100	TCP	74 80 → 53062 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM TSval=4294952466 TSecr=810535439 WS=64
37 36.775803786 192.168.200.100	192.168.200.150	TCP	66 55656 → 22 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=810535439 TSecr=4294952466
38 36.775813232 192.168.200.100	192.168.200.150	TCP	66 53062 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=810535439 TSecr=4294952466
39 36.775861964 192.168.200.100	192.168.200.150	TCP	66 41182 → 21 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=810535439 TSecr=4294952466
40 36.775975876 192.168.200.100	192.168.200.150	TCP	66 55656 → 22 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=810535439 TSecr=4294952466
41 36.776005853 192.168.200.100	192.168.200.150	TCP	66 53062 → 80 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=810535439 TSecr=4294952466

## Perché il SOAR interverrebbe in automatico?

Un SOAR (Security Orchestration, Automation and Response) è un sistema che automatizza la risposta agli eventi di sicurezza. Il SOAR può utilizzare un algoritmo per rilevare l'attacco, ad esempio analizzando il traffico di rete o i log di sistema. Una volta rilevato l'attacco, il SOAR può eseguire azioni automatiche, come bloccare l'IP dell'attaccante o avvisare un team di sicurezza.

## Perché è importante utilizzare un SIEM o un SOAR?

Un SIEM (Security Information and Event Management) è un sistema che raccoglie e analizza i dati di sicurezza. Il SIEM può aiutare a identificare e rispondere agli attacchi in diversi modi, tra cui:

- Identificazione di anomalie: Il SIEM può identificare anomalie nel traffico di rete o nei log di sistema, che potrebbero indicare un attacco.
- Rilevamento di minacce conosciute: Il SIEM può utilizzare liste di minacce conosciute per rilevare attacchi noti.
- Analisi di correlazione: Il SIEM può correlare eventi di sicurezza diversi per identificare attacchi complessi.

Un SOAR può integrare un SIEM per automatizzare la risposta agli eventi di sicurezza. Questo può aiutare a migliorare la rapidità e l'efficacia della risposta agli attacchi.

Ecco alcuni esempi di come un SOAR può essere utilizzato per rispondere automaticamente agli attacchi:

- Blocco dell'IP dell'attaccante: Il SOAR può bloccare l'IP dell'attaccante utilizzando un firewall o un altro dispositivo di rete.
- Avviso di un team di sicurezza: Il SOAR può avvisare un team di sicurezza dell'attacco.
- Avvio di una procedura di risposta automatizzata: Il SOAR può avviare una procedura di risposta automatizzata che può includere azioni come la rimozione di malware o la modifica delle impostazioni di sicurezza.