
S9-L5

Threat Intelligence & IOC

Emanuele Benedetti | 7 febbraio 2025

Consegna

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:

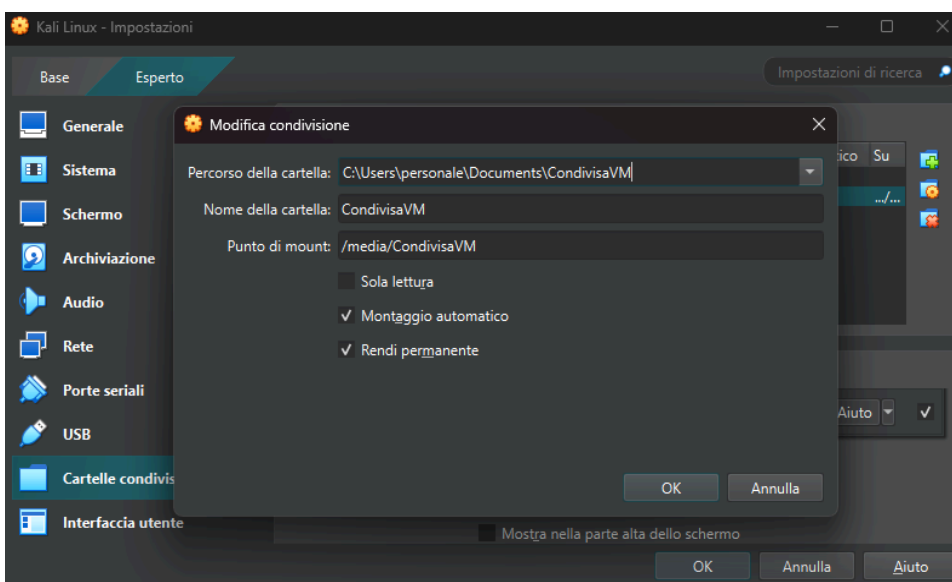
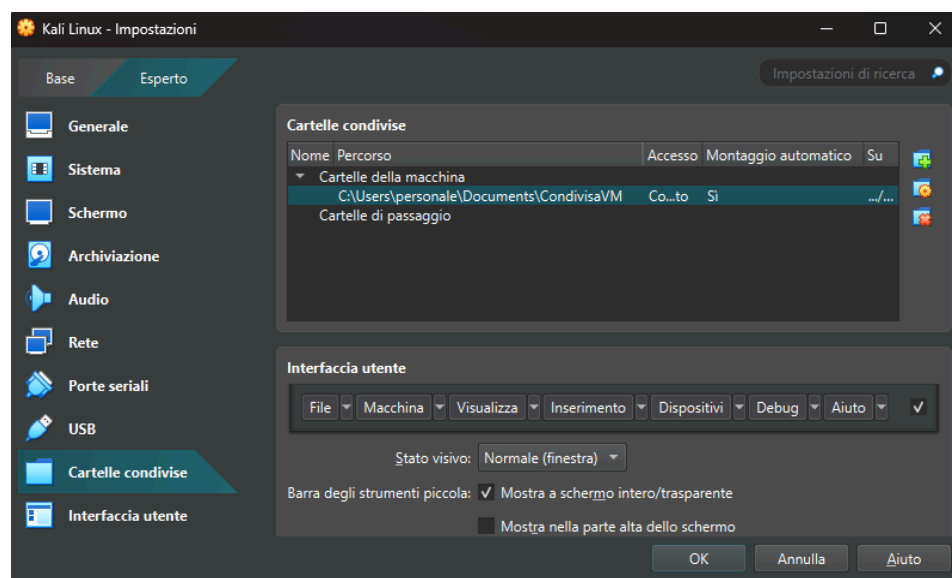
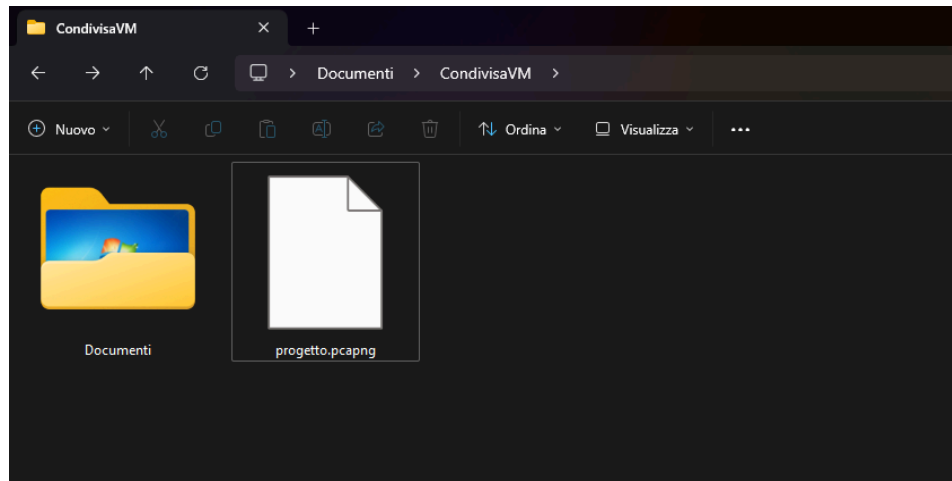
- Identificare ed analizzare eventuali IOC, ovvero evidenze di attacchi in corso
- In base agli IOC trovati, fate delle ipotesi sui potenziali vettori di attacco utilizzati
- Consigliate un'azione per ridurre gli impatti dell'attacco attuale ed eventualmente un simile attacco futuro

Svolgimento

Ho iniziato il laboratorio scaricando il file *progetto.pcapng* fornito dalla consegna.

Creazione cartella condivisa VM

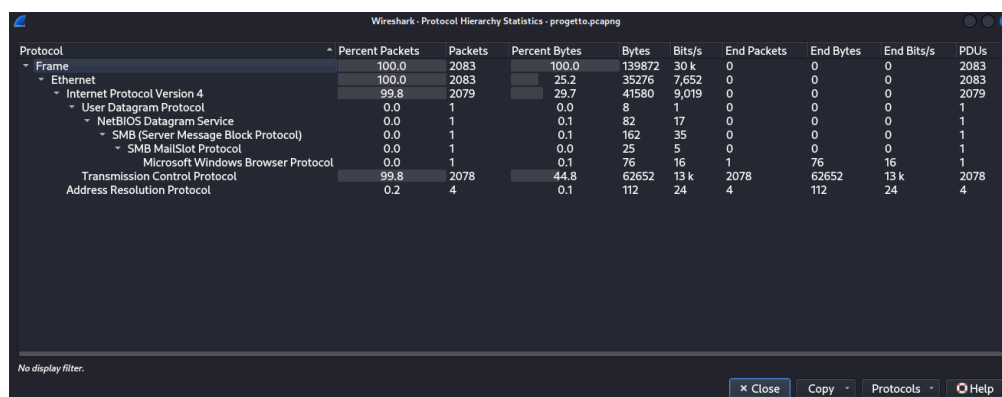
Ho creato una cartella sulla macchina host e condivisa tramite VirtualBox per passare il file alla macchina virtuale Kali Linux, che userò per analizzare la cattura tramite Wireshark. Di seguito sono riportati gli screenshot dei passaggi per la condivisione della cartella tra la macchina host e la macchina virtuale Kali.



Analisi del traffico con Wireshark

Dopo aver importato il file contenente il traffico da analizzare, l'ho aperto con Wireshark per riuscire ad ottenere tutte le informazioni.

Per prima cosa ho controllato le statistiche sui protocolli di rete per avere un'idea dei principali protocolli in gioco in questa analisi. Per fare ciò basta andare in Wireshark > *Statistics* > *Protocol Hierarchy*.



Wireshark - Protocol Hierarchy Statistics - progetto.pcapng

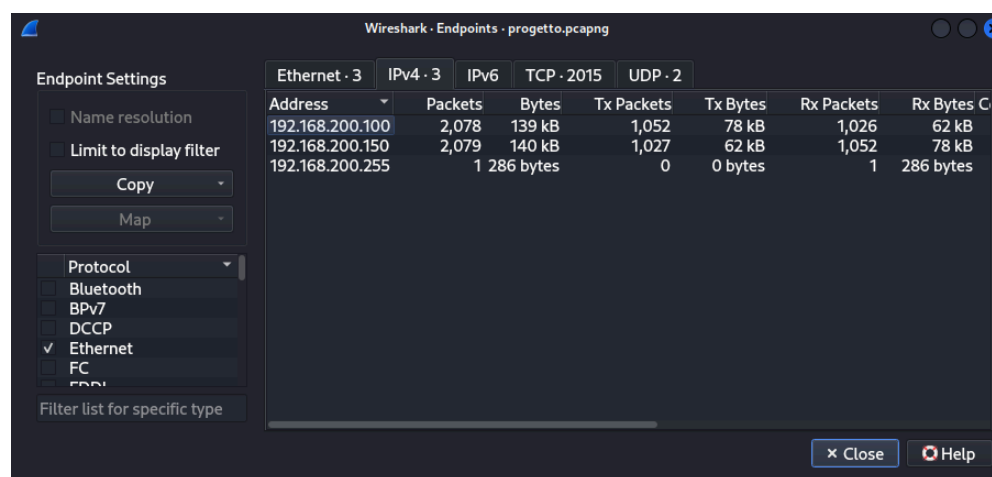
Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s	PDUs
Frame	100.0	2083	100.0	139872	30 k	0	0	0	2083
Ethernet	100.0	2083	25.2	35276	7,652	0	0	0	2083
Internet Protocol Version 4	99.8	2079	29.7	41580	9,019	0	0	0	2079
User Datagram Protocol	0.0	1	0.0	8	1	0	0	0	1
NetBIOS Datagram Service	0.0	1	0.1	82	17	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	112	24	4	112	24	4

No display filter.

Close Copy Protocols Help

Quello che emerge da queste statistiche è che quasi la totalità del traffico (99,8%) avviene su TCP.

Ho inoltre usato le statistiche sugli indirizzi IP per vedere quali fossero i principali attori delle comunicazioni nella cattura del traffico. In questo caso mi sono recato nella sezione *Statistics* > *Endpoints*.



Wireshark - Endpoints - progetto.pcapng

Endpoint Settings

- ☐ Name resolution
- ☐ Limit to display filter
- Copy
- Map
- Protocol: Bluetooth, BPv7, DCCP, ☒ Ethernet, FC, ...
- Filter list for specific type

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

Da questi dati è molto facile vedere che ci sono solamente due dispositivi coinvolti, identificati dagli indirizzi IPv4 *192.168.200.100* e *192.168.200.150*.

Sono passato quindi all'analisi effettiva del traffico. Ho inizialmente dato una rapida lettura ai pacchetti per avere un'idea generale di che tipo di traffico TCP sia stato generato dalle macchine.

The screenshot shows the Wireshark interface with a packet list on the left and a packet details pane on the right. The packet list shows a series of TCP packets between 192.168.200.100 and 192.168.200.150. The packet details pane shows the structure of a selected packet, including Ethernet II, Internet Protocol Version 4, and Transmission Control Protocol fields.

No.	Time	Source	Destination	Protocol	Length	SPort	DPort	Info
1	0.000000	192.168.200.150	192.168.200.100	BROWSER	284			Host Announcement: METASPOITABLE, Workstation, Server, Print, Xenix, NT...
2	2.2376421495	192.168.200.100	192.168.200.150	TCP	74	53600	80	53600 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=810522427 TSecr=0 W...
3	23.764287789	192.168.200.100	192.168.200.150	TCP	74	33876	443	33876 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=810522428 TSecr=0 W...
4	23.764777323	192.168.200.150	192.168.200.100	TCP	74	80	53600	80 → 53600 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM TSval=4294951165 TSecr=0 W...
5	23.764777323	192.168.200.150	192.168.200.100	TCP	60	413	53076	413 → 53076 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
6	23.764515289	192.168.200.100	192.168.200.150	TCP	66	53600	80	53600 → 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	53600	80	53600 → 80 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=810522428 TSecr=4294951165
8	28.761629461	PCSSystemtec_39:7d...	PCSSystemtec_39:7d...	ARP	60			Who has 192.168.200.100? Tell 192.168.200.150
9	28.761644619	PCSSystemtec_39:7d...	PCSSystemtec_39:7d...	ARP	42			Who has 192.168.200.150? Tell 192.168.200.100
10	28.774852257	PCSSystemtec_39:7d...	PCSSystemtec_39:7d...	ARP	42			Who has 192.168.200.150? Tell 192.168.200.100
11	28.775230899	PCSSystemtec_39:7d...	PCSSystemtec_39:7d...	ARP	60			192.168.200.150 is at 08:00:27:fd:87:1e
12	36.774113445	192.168.200.100	192.168.200.150	TCP	74	41304	23	41304 → 23 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=810535437 TSecr=0 W...
13	36.774218116	192.168.200.100	192.168.200.150	TCP	74	56120	111	56120 → 111 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=810535437 TSecr=0 W...
14	36.774257841	192.168.200.100	192.168.200.150	TCP	74	33878	443	33878 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=810535437 TSecr=0 W...
15	36.774366395	192.168.200.100	192.168.200.150	TCP	74	58636	954	58636 → 954 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=810535438 TSecr=0 W...
16	36.774409627	192.168.200.100	192.168.200.150	TCP	74	52358	135	52358 → 135 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=810535438 TSecr=0 W...
17	36.774535534	192.168.200.100	192.168.200.150	TCP	74	46138	993	46138 → 993 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=810535438 TSecr=0 W...
18	36.774614776	192.168.200.100	192.168.200.150	TCP	74	41182	21	41182 → 21 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=810535438 TSecr=0 W...
19	36.774685505	192.168.200.150	192.168.200.100	TCP	74	23	41304	23 → 41304 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM TSval=4294952466 TSecr=0 W...
20	36.774685552	192.168.200.150	192.168.200.100	TCP	74	111	56120	111 → 56120 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM TSval=4294952466 TSecr=0 W...
21	36.774685696	192.168.200.150	192.168.200.100	TCP	60	443	33878	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	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	135 → 52358 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
24	36.774709464	192.168.200.100	192.168.200.150	TCP	66	41304	23	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	56120 → 111 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=810535438 TSecr=4294952466
26	36.775111104	192.168.200.150	192.168.200.100	TCP	60	993	46138	993 → 46138 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
27	36.775111171	192.168.200.150	192.168.200.100	TCP	74	21	41182	21 → 41182 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM TSval=4294952466 TSecr=0 W...
28	36.775174048	192.168.200.100	192.168.200.150	TCP	66	41182	21	41182 → 21 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=810535438 TSecr=4294952466
29	36.775337880	192.168.200.100	192.168.200.150	TCP	74	59174	113	59174 → 113 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=810535438 TSecr=0 W...
30	36.775386694	192.168.200.100	192.168.200.150	TCP	74	55656	22	55656 → 22 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=810535439 TSecr=0 W...
31	36.775524204	192.168.200.100	192.168.200.150	TCP	74	53062	80	53062 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=810535439 TSecr=0 W...
32	36.775589806	192.168.200.150	192.168.200.100	TCP	60	113	59174	113 → 59174 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
33	36.775619454	192.168.200.150	192.168.200.100	TCP	66	41304	23	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	60	56120	111	56120 → 111 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=810535439 TSecr=4294952466
35	36.775796388	192.168.200.150	192.168.200.100	TCP	74	22	55656	22 → 55656 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM TSval=4294952466 TSecr=0 W...
36	36.775797084	192.168.200.150	192.168.200.100	TCP	74	80	53062	80 → 53062 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM TSval=4294952466 TSecr=0 W...
37	36.775893786	192.168.200.100	192.168.200.150	TCP	66	55656	22	55656 → 22 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=810535439 TSecr=4294952466

OS Minor Version: 9
Server Type: 0x00019a03, Workstation, Server, Print, Xenix, NT Workstation, NT Server, Potential Browser
Browser Protocol Major Version: 15
Browser Protocol Minor Version: 1
Signature: 0xaa55
Host Comment: metasploitable server (Samba 3.0.20-Debian)

Packets: 2083 Profile: Default

Sono riuscito ad identificare la macchina con indirizzo IP *192.168.200.100* come macchina attaccante e *192.168.200.150* come vittima dell'attacco. Ad inizio cattura inoltre la macchina vittima si “presenta” come una macchina Metasploitable, come è possibile notare dalla sezione *Info* dello screenshot.

Analisi generale della cattura del traffico

In generale appare evidente che ciò che sta avvenendo è che l'attaccante sta sfruttando l'handshake a tre vie del protocollo TCP. Vengono aperte infatti numerose connessioni con il target, su vari servizi e subito dopo la ricezione del pacchetto ACK da parte della vittima, spesso l'attaccante invia un pacchetto *RST* che ha la funzione di terminare immediatamente la connessione, prima di inviare informazioni utili.

Possiamo così riassumere ciò che sta accadendo:

1. Fase di connessione

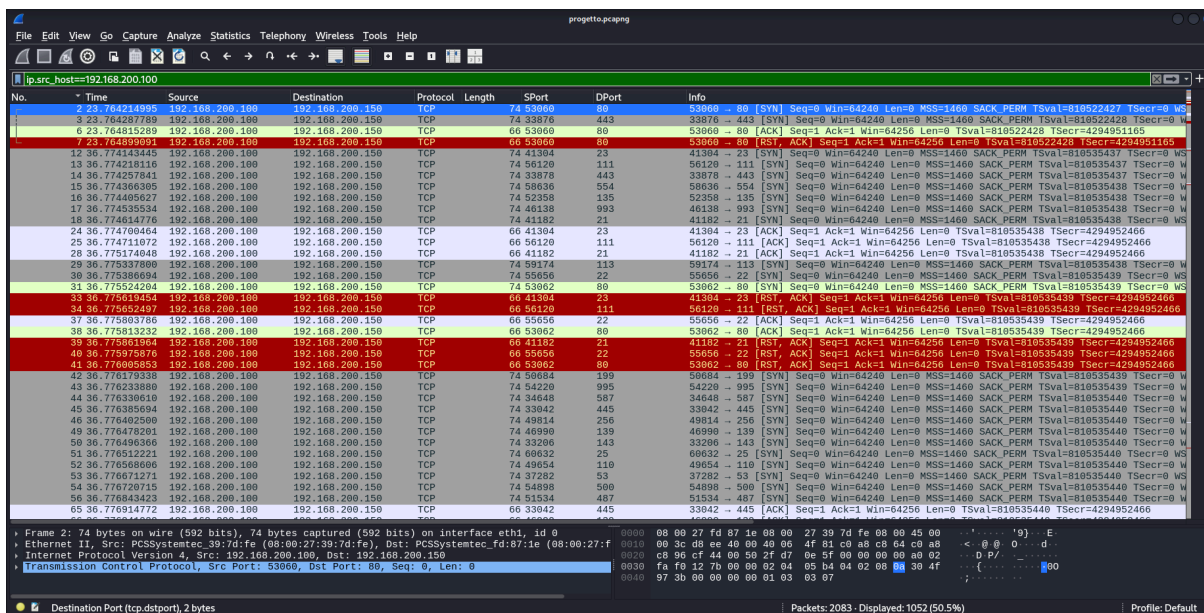
- L'IP 192.168.200.100 (attaccante) invia pacchetti SYN a più porte su 192.168.200.150 (vittima)
- La vittima risponde con il pacchetto SYN-ACK
- L'attaccante completa l'handshake con il pacchetto ACK

2. Fase di reset immediato

- Dopo aver completato l'handshake l'attaccante invia un pacchetto RST, ACK che interrompe immediatamente la connessione appena avviata

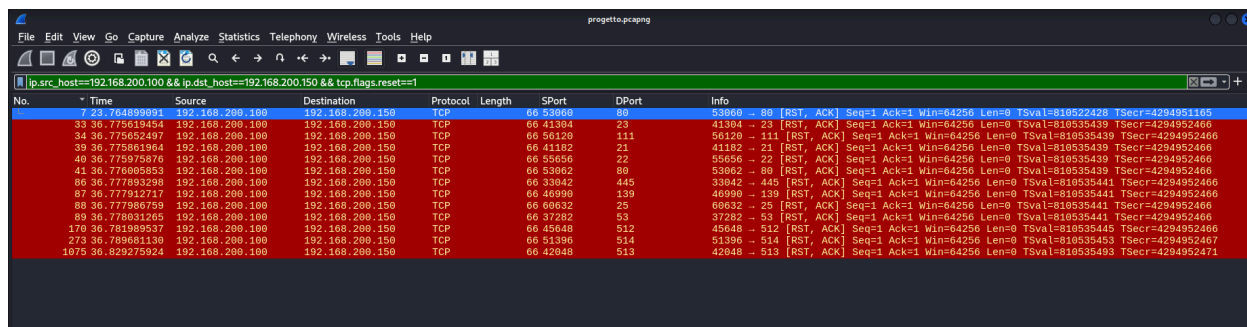
Analisi dei pacchetti con i filtri

Ho utilizzato dei filtri in Wireshark per ottenere solamente una visuale dei pacchetti inviati dalla macchina attaccante tramite il comando *ip.src_host=192.168.200.100* in modo tale da vedere quali connessioni ha tentato di stabilire e come le ha gestite.



Lo screenshot mostra quanto precedentemente spiegato, possiamo apprezzare infatti la quantità di pacchetti SYN inviati dall'attaccante verso le porte della macchina target.

Ho quindi modificato il filtro inserendo il comando `ip.src_host=192.168.200.100 && ip.dst_host==192.168.200.150 && tcp.flags.reset==1` per vedere solamente i pacchetti RST inviati dalla macchina attaccante per chiudere le connessioni avviate con il server.

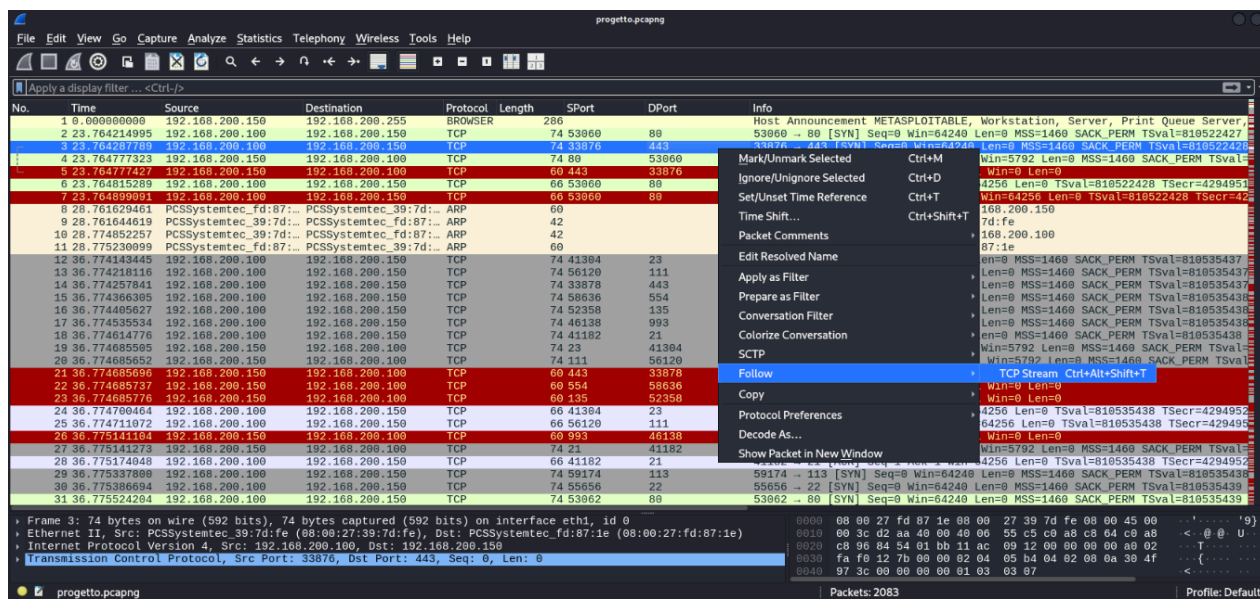


The image shows a Wireshark packet capture with a display filter of `ip.src_host=192.168.200.100 && ip.dst_host==192.168.200.150 && tcp.flags.reset==1`. The packet list shows several RST (Reset) packets from the source IP 192.168.200.100 to the destination IP 192.168.200.150. The packets are numbered 23, 33, 39, 40, 41, 86, 87, 88, 89, 170, 173, and 1075. The 'Info' column for each packet shows details like 'Seq=1 Ack=1 Win=64256 Len=0 TSval=810522428 TSecr=4294952466'.

No.	Time	Source	Destination	Protocol	Length	SPort	DPort	Info
23	0.000000	192.168.200.100	192.168.200.150	TCP	66	53960	80	53960 → 80 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=810522428 TSecr=4294952466
33	0.000000	192.168.200.100	192.168.200.150	TCP	66	41304	23	41304 → 23 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=810522428 TSecr=4294952466
39	0.000000	192.168.200.100	192.168.200.150	TCP	66	56120	111	56120 → 111 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=810522428 TSecr=4294952466
40	0.000000	192.168.200.100	192.168.200.150	TCP	66	41102	21	41102 → 21 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=810522428 TSecr=4294952466
41	0.000000	192.168.200.100	192.168.200.150	TCP	66	55656	22	55656 → 22 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=810522428 TSecr=4294952466
86	0.000000	192.168.200.100	192.168.200.150	TCP	66	53062	80	53062 → 80 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=810522428 TSecr=4294952466
87	0.000000	192.168.200.100	192.168.200.150	TCP	66	33042	445	33042 → 445 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=810522428 TSecr=4294952466
88	0.000000	192.168.200.100	192.168.200.150	TCP	66	46990	139	46990 → 139 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=810522428 TSecr=4294952466
89	0.000000	192.168.200.100	192.168.200.150	TCP	66	60632	25	60632 → 25 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=810522428 TSecr=4294952466
170	0.000000	192.168.200.100	192.168.200.150	TCP	66	37282	53	37282 → 53 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=810522428 TSecr=4294952466
173	0.000000	192.168.200.100	192.168.200.150	TCP	66	45648	512	45648 → 512 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=810522428 TSecr=4294952466
1075	0.000000	192.168.200.100	192.168.200.150	TCP	66	51396	514	51396 → 514 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=810522428 TSecr=4294952466

L'immagine mostra che vengono aperte ed immediatamente chiuse le connessioni con le porte 80, 23, 111, 21, 22, 445, 139, 25, 53, 512, 514, 513.

Per finire possiamo utilizzare la funzione per seguire lo stream TCP cliccando un pacchetto di interesse con il tasto destro > *Follow* > *TCP Stream*



The image shows a Wireshark packet capture with a display filter of `ip.src_host=192.168.200.100 && ip.dst_host==192.168.200.150 && tcp.flags.reset==1`. The packet list shows several RST (Reset) packets from the source IP 192.168.200.100 to the destination IP 192.168.200.150. The packets are numbered 23, 33, 39, 40, 41, 86, 87, 88, 89, 170, 173, and 1075. The 'Info' column for each packet shows details like 'Seq=1 Ack=1 Win=64256 Len=0 TSval=810522428 TSecr=4294952466'. A right-click context menu is open over packet 23, and the 'Follow' > 'TCP Stream' option is selected.

No.	Time	Source	Destination	Protocol	Length	SPort	DPort	Info
23	0.000000	192.168.200.100	192.168.200.150	TCP	66	53960	80	53960 → 80 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=810522428 TSecr=4294952466
33	0.000000	192.168.200.100	192.168.200.150	TCP	66	41304	23	41304 → 23 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=810522428 TSecr=4294952466
39	0.000000	192.168.200.100	192.168.200.150	TCP	66	56120	111	56120 → 111 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=810522428 TSecr=4294952466
40	0.000000	192.168.200.100	192.168.200.150	TCP	66	41102	21	41102 → 21 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=810522428 TSecr=4294952466
41	0.000000	192.168.200.100	192.168.200.150	TCP	66	55656	22	55656 → 22 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=810522428 TSecr=4294952466
86	0.000000	192.168.200.100	192.168.200.150	TCP	66	53062	80	53062 → 80 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=810522428 TSecr=4294952466
87	0.000000	192.168.200.100	192.168.200.150	TCP	66	33042	445	33042 → 445 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=810522428 TSecr=4294952466
88	0.000000	192.168.200.100	192.168.200.150	TCP	66	46990	139	46990 → 139 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=810522428 TSecr=4294952466
89	0.000000	192.168.200.100	192.168.200.150	TCP	66	60632	25	60632 → 25 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=810522428 TSecr=4294952466
170	0.000000	192.168.200.100	192.168.200.150	TCP	66	37282	53	37282 → 53 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=810522428 TSecr=4294952466
173	0.000000	192.168.200.100	192.168.200.150	TCP	66	45648	512	45648 → 512 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=810522428 TSecr=4294952466
1075	0.000000	192.168.200.100	192.168.200.150	TCP	66	51396	514	51396 → 514 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=810522428 TSecr=4294952466

In questo modo possiamo vedere il flusso TCP per ogni connessione stabilita e successivamente resettata.

Viene quindi aperta una nuova finestra di Wireshark che normalmente contiene i dati e le informazioni scambiate tra client e server durante la comunicazione ma poiché il pacchetto RST viene inviato senza aver scambiato alcun dato tra le macchine, non è presente alcuna informazione.

Tuttavia il filtro per seguire lo stream ci è molto utile poiché vengono filtrati singolarmente i pacchetti della connessione stabilita e resettata. Ci basta incrementare il numero dello stream per vedere tutte quante le connessioni al variare della porta.

Nelle immagini di seguito mostro alcuni degli stream aperti e chiusi dall'attaccante:

Stream 0 - Porta 80 HTTP:

No.	Time	Source	Destination	Protocol	Length	SPort	DPort	Info
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_P
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=1
6	23.764815289	192.168.200.100	192.168.200.150	TCP	66	53060	80	53060 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=810
7	23.764899091	192.168.200.100	192.168.200.150	TCP	66	53060	80	53060 → 80 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSva

Stream 2 - Porta 23 Telnet:

No.	Time	Source	Destination	Protocol	Length	SPort	DPort	Info
12	36.774143445	192.168.200.100	192.168.200.150	TCP	74	41304	23	41304 → 23 [SYN, Seq=0 Win=64240 Len=0 MSS=1460 SACK_P
19	36.774685595	192.168.200.150	192.168.200.100	TCP	74	23	41304	23 → 41304 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1
24	36.774708464	192.168.200.100	192.168.200.150	TCP	66	41304	23	41304 → 23 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=810
33	36.775619454	192.168.200.100	192.168.200.150	TCP	66	41304	23	41304 → 23 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSva

Stream 8 - Porta 21 FTP:

No.	Time	Source	Destination	Protocol	Length	SPort	DPort	Info
18	36.774614776	192.168.200.100	192.168.200.150	TCP	74	41182	21	41182 → 21 [SYN, Seq=0 Win=64240 Len=0 MSS=1460 SACK_P
27	36.775141273	192.168.200.150	192.168.200.100	TCP	74	21	41182	21 → 41182 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1
28	36.775174048	192.168.200.100	192.168.200.150	TCP	66	41182	21	41182 → 21 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=810
39	36.775861964	192.168.200.100	192.168.200.150	TCP	66	41182	21	41182 → 21 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSva

Identificazione degli IOC

La tabella di seguito riassume gli indicatori di compromissione del possibile attacco:

IOC	Descrizione
Fonte dell'attacco	IP attaccante: 192.168.200.100
Target	IP vittima: 192.168.200.150
Tecnica usata	Invio pacchetti TCP RST
Caratteristiche	Handshake TCP e invio di pacchetti RST immediati
Possibili obiettivi	Scansione delle porte, attacco DoS (RST flood)

Ipotesi sui potenziali vettori di attacco

Ipotesi 1: Scansione delle porte

Sulla base degli indicatori che ho trovato, ho ipotizzato che l'attaccante abbia eseguito una scansione stealth ad esempio con *nmap* sulle porte TCP del target. L'ipotesi della scansione trova riscontro nella quantità di pacchetti SYN inviati dall'attaccante e nei pacchetti RST inviati dal server al client dopo il primo pacchetto SYN ricevuto, il che indica che la porta scansionata è chiusa.

Ipotesi 2: Attacco DoS - RST flood

Considerando invece le connessioni terminate immediatamente dall'attaccante, possiamo ipotizzare che sia in corso un attacco DoS di tipo RST flood.

L'attacco RST flood è un attacco DoS che rientra negli attacchi di esaurimento di stato. Possiamo spiegare in breve l'attacco come segue: quando viene avviata una connessione tra client e server, quest'ultimo riserva una quantità di risorse per la connessione. Poiché la capacità di risorse di un dispositivo è limitata, gli attacchi ad esaurimento di stato mirano a saturare questa capacità inviando numerose richieste di connessione. L'attaccante invia tanti pacchetti RST con l'obiettivo di

chiudere le connessioni in modo ripetuto, sovraccaricando il sistema e riducendo la disponibilità del servizio.

L'attacco RST flood è stato usato ad esempio nel 2004 con una serie di attacchi coordinati verso diversi router di ISP compromettendo la stabilità di diverse reti in Nord America e in Europa. Gli attaccanti miravano ad interrompere le connessioni tra i principali provider di backbone (dorsali che gestiscono il traffico principale tra i nodi della rete).

Possibili contromisure

Di seguito ho elencato possibili contromisure per ridurre l'impatto di questo attacco e prevenire attacchi simili in futuro:

1. Contromisure immediate

- Bloccare l'indirizzo IP dell'attaccante (ad esempio tramite il firewall)
- Usare tool di rilevamento di pacchetti RST sospetti (ad esempio Snort)
- Limitare i pacchetti RST per evitare il flooding

2. Protezione a lungo termine e prevenzione

- Abilitare il logging avanzato per rilevare pacchetti RST anomali
- Impedire agli utenti non autorizzati di effettuare scanning della rete
- Segmentare la rete per limitare movimenti laterali

Conclusione

Dall'analisi del traffico di rete è possibile ipotizzare che è in corso una ricognizione e forse un attacco DoS.

L'indirizzo IP della macchina dell'attaccante è 192.168.200.100. Con quasi certezza vi è una scansione delle porte sulla macchina target, per i numerosi tentativi di connessione alle porte e un probabile attacco DoS di tipo RST flood, indicato dal numero di connessioni terminate immediatamente dall'attaccante.

Bonus

Consegna

Siete chiamati a progettare le difese di questo scenario:

Azienda Mak produce dei macchinari e il cliente vuole mettere in sicurezza tutto l'ecosistema. Abbiamo da una parte l'azienda Mak, poi c'è il macchinario e dall'altra parte c'è il cliente che lo utilizza.

Il macchinario è basato su Windows 10, ha porta di rete (usata solo per gli aggiornamenti e la diagnostica remota), porta USB (sono disabilitate le pendrive, ovviamente)

La diagnostica remota è fatta attraverso la VPN del cliente

Il macchinario è sostanzialmente bloccato – La partizione del sistema operativo non è scrivibile mentre c'è una seconda partizione per il software di gestione del macchinario. Il software di gestione è realizzato con il linguaggio C99.

Il macchinario è installato nelle varie aziende clienti.

La consegna richiede di:

1. Valutare le eventuali vulnerabilità e punti di attacco
1. Proporre al cliente soluzioni di sicurezza
2. Progettare un sistema di monitoraggio del traffico (Windows 10 è bloccato dalla casa madre, non è modificabile)

Proporre al cliente due soluzioni, una economica (massimo 500 euro) e una più costosa (massimo 2500 euro)

Svolgimento

Valutazione delle vulnerabilità e punti di Attacco

Sulla base delle informazioni fornite ho identificato le seguenti vulnerabilità:

1. Attacchi alla rete

- Se la VPN non è configurata correttamente, la connessione potrebbe essere compromessa
- Traffico di rete non cifrato o monitorato potrebbe essere intercettato
- Attacchi di tipo *Man-in-the-Middle* durante la diagnostica remota

2. Attacchi al software di gestione

- Il software in C99 potrebbe contenere vulnerabilità come buffer overflow, use-after-free, o altre tipiche di linguaggi a basso livello
- Se la partizione del software di gestione è scrivibile, potrebbe essere compromessa da malware o exploit

3. Attacchi fisici

- Accesso non autorizzato alla porta USB potrebbe consentire l'inserimento di dispositivi malevoli (es. tastiere con payload malevolo).
- Furto o manipolazione fisica del macchinario

4. Aggiornamenti insicuri

- Gli aggiornamenti del sistema o del software di gestione potrebbero essere veicoli per malware se non verificati con firme digitali

5. Attacchi interni

- Se il cliente ha una rete non sicura, il macchinario potrebbe essere esposto a minacce provenienti dall'interno della rete locale

Proposte di soluzione di sicurezza

Considerate le minacce e le vulnerabilità appena elencate propongo la seguente lista di soluzioni di sicurezza per ogni ambito analizzato

1. Sicurezza della rete

- Configurare la VPN con protocolli sicuri (IPsec, OpenVPN o Wireguard)
- Abilitare l'autenticazione a due fattori (2FA) per l'accesso alla VPN
- Isolare il macchinario in una VLAN dedicata per limitare l'esposizione alla rete locale del cliente
- Implementare un firewall hardware o software per bloccare tutto il traffico non necessario

2. Protezione del software di gestione

- Eseguire regolarmente test di sicurezza sul software in C99 per identificare e correggere vulnerabilità
- Utilizzare meccanismi di firma digitale per verificare l'integrità degli aggiornamenti del software

3. Protezione fisica

- Installare custodie protettive per coprire le porte USB e di rete, accessibili solo con chiavi speciali
- Monitorare gli accessi fisici al macchinario tramite telecamere o sistemi di allarme

4. Monitoraggio e logging

- Implementare un sistema di logging centralizzato per raccogliere e analizzare gli eventi di sicurezza dal macchinario
- Configurare avvisi automatici per attività sospette (ad esempio tentativi di accesso non autorizzati)

Progettazione di un sistema di monitoraggio del traffico

Poiché Windows 10 è bloccato dalla casa madre, il monitoraggio deve essere implementato esternamente al macchinario.

Ho creato due soluzioni, una economica (massimo 500€) e una più avanzata (massimo 2500€), come richiesto dalla consegna

- 1. Soluzione economica:** l'obiettivo è implementare un sistema di monitoraggio di base che consenta di raccogliere e analizzare il traffico di rete generato dal macchinario, senza investire in hardware o software costosi. Le componenti necessarie sono:
 - PC usato di fascia medio-bassa con 2 schede di rete (circa 300€)
 - Eventuale scheda di rete aggiuntiva, se il PC ne possiede una sola (25€)
 - Switch gestito per duplicare il traffico di rete verso il sistema di monitoraggio (circa 90€)
 - Software per il monitoraggio del traffico (gratuito)
 - Wireshark per l'analisi del traffico di rete
 - Zeek per il monitoraggio avanzato del traffico e la rilevazione di anomalie
 - Disco esterno per archiviare i log del traffico (circa 80€)
- 2. Soluzione avanzata:** l'obiettivo è implementare un sistema di monitoraggio avanzato che fornisca funzionalità di IDS/IPS e integrazione con un SIEM per una gestione centralizzata dei log. Le componenti necessarie sono:
 - Firewall avanzato con IDS/IPS (circa 1200€)
 - Server dedicato per il SIEM (circa 800€)
 - Switch con mirroring per inviare copie del traffico al sistema di monitoraggio (circa 300€)
 - Software di gestione SIEM come Wazuh o ELK Stack (gratuito)
 - Disco esterno per archiviare i log del traffico (circa 100€)

Ho inoltre creato una tabella di confronto delle due soluzioni

Caratteristica	Soluzione economica (495€)	Soluzione avanzata (2400€)
Hardware principale	PC usato	Server dedicato con firewall
Software	Wireshark/Zeek	Wazuh/ELK Stack
Funzionalità IDS/IPS	No	Si (nel firewall)
Monitoraggio centralizzato	No	Si (SIEM)
Archiviazione log	Si (disco esterno)	Si (disco esterno)
Complessità di configurazione	Bassa	Alta

La soluzione economica è ideale per clienti con budget limitato che richiedono un monitoraggio di base. È sufficiente per identificare anomalie di rete e raccogliere log, ma manca di funzionalità avanzate come IDS/IPS e monitoraggio centralizzato.

La soluzione avanzata è consigliata per clienti che desiderano un livello superiore di sicurezza e monitoraggio. Offre funzionalità avanzate come IDS/IPS, SIEM e avvisi automatici, garantendo una protezione più robusta.