
Progetto L9/s5

07/02/25

CONSEGNA

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. Esercizio Threat Intelligence & IOC 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

FILE ANALIZZATO

link:

<https://mega.nz/file/MTA1xJgS#NOwoJs234TUkNV4o4Pv5d0DoRPeTVszlm8-QOYrV-4I>

SPECIFICHE

Il file analizzato contiene un'analisi effettuata con Wireshark che mostra una serie molto lunga di richieste e trasmissioni di dati tra l'indirizzo IP

192.168.200.100

e

192.168.200.150

il che ci fa intendere, tanto per cominciare, che entrambi gli apparecchi sono posizionati nello

dettaglio che potrà tornarci utile più avanti, ma andiamo con ordine.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.200.150	192.168.200.255	BROADCAST	288	Host Announcement METASPLOITABLE, Workstation, Server, Print Queue Server, Xenix Server, NT Workstation, NT Server, Potential
2	0.23.764214995	192.168.200.100	192.168.200.150	TCP	74	53060 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 Tsvail=810522427 TSecr=WS=128
3	0.23.764287789	192.168.200.100	192.168.200.150	TCP	74	33076 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 Tsvail=810522428 TSecr=WS=128
4	0.23.764777323	192.168.200.150	192.168.200.100	TCP	74	88 → 53060 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM=1 Tsvail=4294951165 TSecr=810522427 WS=64
5	0.23.764777323	192.168.200.150	192.168.200.100	TCP	74	88 → 53076 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM=1 Tsvail=4294951165 TSecr=810522427 WS=64
6	0.23.764815020	192.168.200.100	192.168.200.150	TCP	66	53060 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 Tsvail=810522428 TSecr=4294951165
7	0.23.764899091	192.168.200.100	192.168.200.150	TCP	66	53060 → 80 (RST, ACK) Seq=1 Ack=1 Win=64256 Len=0 Tsvail=810522428 TSecr=4294951165
8	0.28.761629641	PcsCompu_39:7d:fe	PcsCompu_39:7d:fe	ARP	60	Who has 192.168.200.100 Tell 192.168.200.150
9	0.28.761644619	PcsCompu_39:7d:fe	PcsCompu_39:7d:fe	ARP	42	192.168.200.100 is at 08:00:27:39:7d:fe
10	0.27.764852255	192.168.200.100	192.168.200.150	ARP	42	Who has 192.168.200.150 Tell 192.168.200.100
11	0.28.775230909	PcsCompu_39:7d:fe	PcsCompu_39:7d:fe	ARP	60	192.168.200.150 is at 08:00:27:7d:fe:87:1e
12	0.26.774143445	192.168.200.100	192.168.200.150	TCP	74	41304 → 23 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 Tsvail=810535437 TSecr=WS=128
13	0.26.774258116	192.168.200.100	192.168.200.150	TCP	74	56120 → 111 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 Tsvail=810535437 TSecr=WS=128
14	0.26.774258116	192.168.200.100	192.168.200.150	TCP	74	33078 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 Tsvail=810535437 TSecr=WS=128
15	0.26.774363030	192.168.200.100	192.168.200.150	TCP	74	53060 → 554 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 Tsvail=810535438 TSecr=WS=128
16	0.26.774405627	192.168.200.100	192.168.200.150	TCP	74	52358 → 135 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 Tsvail=810535438 TSecr=WS=128
17	0.26.774535534	192.168.200.100	192.168.200.150	TCP	74	46138 → 993 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 Tsvail=810535438 TSecr=WS=128
18	0.26.774614776	192.168.200.100	192.168.200.150	TCP	74	41182 → 21 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 Tsvail=810535438 TSecr=WS=128
19	0.26.774685505	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=1 Tsvail=4294952466 TSecr=810535437 WS=64
20	0.26.774685505	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=1 Tsvail=4294952466 TSecr=810535437 WS=64
21	0.26.774685996	192.168.200.150	192.168.200.100	TCP	60	443 → 33078 (RST, ACK) Seq=1 Ack=1 Win=0 Len=0
22	0.26.774685737	192.168.200.150	192.168.200.100	TCP	60	554 → 58636 (RST, ACK) Seq=1 Ack=1 Win=0 Len=0
23	0.26.774685776	192.168.200.150	192.168.200.100	TCP	60	135 → 52358 (RST, ACK) Seq=1 Ack=1 Win=0 Len=0
24	0.26.774709644	192.168.200.150	192.168.200.100	TCP	66	41304 → 23 [ACK] Seq=1 Ack=1 Win=64256 Len=0 Tsvail=810535438 TSecr=4294952466
25	0.26.774711973	192.168.200.150	192.168.200.100	TCP	66	56120 → 111 [ACK] Seq=1 Ack=1 Win=64256 Len=0 Tsvail=810535438 TSecr=4294952466
26	0.26.775141104	192.168.200.150	192.168.200.100	TCP	60	993 → 46138 (RST, ACK) Seq=1 Ack=1 Win=0 Len=0
27	0.26.775141273	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=1 Tsvail=4294952466 TSecr=810535438 WS=64

Il protocollo utilizzato permette di **mappare la rete e identificare asset vulnerabili**, spesso questo è indice di una scansione iniziale che può portare ad identificare target vulnerabili e può condurre poi a uno o più attacchi di varia natura

2 Presenza di numerosa richieste SYN senza HACK

No.	Time	Source	Destination	Protocol	Length	Info
2	23.764214995	192.168.200.100	192.168.200.150	TCP	74	53668 → 88 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=810522427 TSecr=0 WS=128
3	23.764287789	192.168.200.100	192.168.200.150	TCP	74	33976 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=810522428 TSecr=0 WS=128
12	36.774113145	192.168.200.100	192.168.200.150	TCP	74	41394 → 23 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=810535437 TSecr=0 WS=128
13	36.774218116	192.168.200.100	192.168.200.150	TCP	74	56129 → 111 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=810535437 TSecr=0 WS=128
14	36.774257841	192.168.200.100	192.168.200.150	TCP	74	33878 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=810535437 TSecr=0 WS=128
15	36.774366395	192.168.200.100	192.168.200.150	TCP	74	58635 → 554 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=810535438 TSecr=0 WS=128
16	36.774405627	192.168.200.100	192.168.200.150	TCP	74	52358 → 135 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=810535438 TSecr=0 WS=128
17	36.774535534	192.168.200.100	192.168.200.150	TCP	74	46138 → 993 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=810535438 TSecr=0 WS=128
18	36.774614776	192.168.200.100	192.168.200.150	TCP	74	41182 → 21 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=810535438 TSecr=0 WS=128
29	36.775337890	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	53662 → 88 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=810535439 TSecr=0 WS=128
42	36.776179338	192.168.200.100	192.168.200.150	TCP	74	58684 → 199 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=810535439 TSecr=0 WS=128
43	36.776233880	192.168.200.100	192.168.200.150	TCP	74	54228 → 995 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=810535439 TSecr=0 WS=128
44	36.776338610	192.168.200.100	192.168.200.150	TCP	74	34648 → 587 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=810535440 TSecr=0 WS=128
45	36.776385694	192.168.200.100	192.168.200.150	TCP	74	33042 → 445 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=810535440 TSecr=0 WS=128
46	36.776402590	192.168.200.100	192.168.200.150	TCP	74	49814 → 256 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=810535440 TSecr=0 WS=128
49	36.776478201	192.168.200.100	192.168.200.150	TCP	74	46998 → 139 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=810535440 TSecr=0 WS=128
50	36.776496360	192.168.200.100	192.168.200.150	TCP	74	33296 → 143 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=810535440 TSecr=0 WS=128
51	36.776512221	192.168.200.100	192.168.200.150	TCP	74	69632 → 25 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=810535440 TSecr=0 WS=128
52	36.776568606	192.168.200.100	192.168.200.150	TCP	74	49654 → 110 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=810535440 TSecr=0 WS=128
53	36.776671271	192.168.200.100	192.168.200.150	TCP	74	37282 → 53 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=810535440 TSecr=0 WS=128
54	36.776728715	192.168.200.100	192.168.200.150	TCP	74	54898 → 580 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=810535440 TSecr=0 WS=128
56	36.776843423	192.168.200.100	192.168.200.150	TCP	74	51534 → 487 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=810535440 TSecr=0 WS=128
70	36.777143014	192.168.200.100	192.168.200.150	TCP	74	56990 → 707 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=810535440 TSecr=0 WS=128
71	36.777186821	192.168.200.100	192.168.200.150	TCP	74	35638 → 436 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=810535440 TSecr=0 WS=128
72	36.777302991	192.168.200.100	192.168.200.150	TCP	74	34120 → 98 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=810535441 TSecr=0 WS=128
73	36.777337934	192.168.200.100	192.168.200.150	TCP	74	49780 → 78 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=810535441 TSecr=0 WS=128
76	36.777473818	192.168.200.100	192.168.200.150	TCP	74	36138 → 580 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=810535441 TSecr=0 WS=128
77	36.777522494	192.168.200.100	192.168.200.150	TCP	74	52428 → 962 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=810535441 TSecr=0 WS=128
80	36.777645027	192.168.200.100	192.168.200.150	TCP	74	41874 → 764 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=810535441 TSecr=0 WS=128
81	36.777688988	192.168.200.100	192.168.200.150	TCP	74	51506 → 435 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=810535441 TSecr=0 WS=128
89	36.778173978	192.168.200.100	192.168.200.150	TCP	74	54450 → 148 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=810535441 TSecr=0 WS=128
91	36.778206161	192.168.200.100	192.168.200.150	TCP	74	48448 → 806 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=810535441 TSecr=0 WS=128
92	36.778307830	192.168.200.100	192.168.200.150	TCP	74	54566 → 221 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=810535442 TSecr=0 WS=128
96	36.778482791	192.168.200.100	192.168.200.150	TCP	74	42420 → 1007 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=810535442 TSecr=0 WS=128
97	36.778591226	192.168.200.100	192.168.200.150	TCP	74	34646 → 206 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=810535442 TSecr=0 WS=128
98	36.778614055	192.168.200.100	192.168.200.150	TCP	74	64292 → 151 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=810535442 TSecr=0 WS=128

Come possiamo notare il nostro “attaccante” o presunto tale, sta tentando di inviare numerosissime richieste **SYN** in rapida successione.

Se volessimo essere cauti a riguardo, potremmo interpretare questo come un tentativo di scanning delle porte sulla macchina target, visto il numero di trasmissioni in pochissimo tempo, tra gli strumenti da noi utilizzati potrebbe trattarsi probabilmente di una scansione effettuata con **nmap**.

eventualmente potrebbe anche trattarsi di un invio massivo di richieste SYN al fine di sovraccaricare la macchina target, ma dovremmo in quel caso notare molte più richieste per ogni singola porta. Questo tipo di attacchi prende il nome di

SYN flood

per visualizzare le richieste nell'immagine soprastante ho utilizzato il filtro

tcp.flags.syn == 1 && tcp.flags.ack == 0

direttamente da wireshark.

3 Notevole quantità di pacchetti TCP (RST)

No.	Time	Source	Destination	Protocol	Length	Info
5	24.764777427	192.168.200.150	192.168.200.100	TCP	60	443 → 33870 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
7	23.764899091	192.168.200.100	192.168.200.150	TCP	60	53060 → 80 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=810522426 TSecr=4294951105
21	36.774685696	192.168.200.150	192.168.200.100	TCP	60	443 → 33870 [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
26	36.775141894	192.168.200.150	192.168.200.100	TCP	60	993 → 46130 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
32	36.775589800	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	60	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 [RST, 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	60	41102 → 21 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=810535439 TSecr=4294952466
40	36.775975070	192.168.200.100	192.168.200.150	TCP	60	55656 → 22 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=810535439 TSecr=4294952466
41	36.776085853	192.168.200.100	192.168.200.150	TCP	60	53062 → 80 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=810535439 TSecr=4294952466
47	36.776451284	192.168.200.150	192.168.200.100	TCP	60	199 → 50684 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
48	36.776451357	192.168.200.150	192.168.200.100	TCP	60	995 → 54220 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
53	36.776813123	192.168.200.150	192.168.200.100	TCP	60	587 → 34640 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
58	36.776984922	192.168.200.150	192.168.200.100	TCP	60	256 → 49814 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
60	36.776985004	192.168.200.150	192.168.200.100	TCP	60	143 → 33206 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
62	36.776985082	192.168.200.150	192.168.200.100	TCP	60	110 → 49654 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
64	36.776985162	192.168.200.150	192.168.200.100	TCP	60	509 → 54890 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
69	36.777118481	192.168.200.150	192.168.200.100	TCP	60	487 → 51534 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
74	36.777430632	192.168.200.150	192.168.200.100	TCP	60	707 → 56990 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
75	36.777430741	192.168.200.150	192.168.200.100	TCP	60	436 → 35638 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
78	36.777623002	192.168.200.150	192.168.200.100	TCP	60	98 → 34120 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
79	36.777623149	192.168.200.150	192.168.200.100	TCP	60	78 → 49780 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
82	36.777758636	192.168.200.150	192.168.200.100	TCP	60	580 → 36138 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
83	36.777758696	192.168.200.150	192.168.200.100	TCP	60	962 → 52428 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
84	36.777871245	192.168.200.150	192.168.200.100	TCP	60	784 → 41874 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
85	36.777871293	192.168.200.150	192.168.200.100	TCP	60	435 → 51500 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
86	36.777893298	192.168.200.100	192.168.200.150	TCP	60	33042 → 445 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=810535441 TSecr=4294952466
87	36.777912717	192.168.200.100	192.168.200.150	TCP	60	46990 → 139 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=810535441 TSecr=4294952466
88	36.777986759	192.168.200.100	192.168.200.150	TCP	60	60632 → 25 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=810535441 TSecr=4294952466
89	36.778031265	192.168.200.100	192.168.200.150	TCP	60	37282 → 53 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=810535441 TSecr=4294952466
93	36.778385846	192.168.200.150	192.168.200.100	TCP	60	148 → 51450 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
94	36.778385948	192.168.200.150	192.168.200.100	TCP	60	886 → 48448 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
95	36.778449494	192.168.200.150	192.168.200.100	TCP	60	221 → 54566 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
99	36.778636864	192.168.200.150	192.168.200.100	TCP	60	1007 → 42490 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
100	36.778721080	192.168.200.150	192.168.200.100	TCP	60	206 → 34646 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
103	36.778826284	192.168.200.150	192.168.200.100	TCP	60	131 → 54202 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

utilizzando il filtro

`tcp.flags.reset == 1 && ip.src == 192.168.200.100`

possiamo notare un anomalo quantitativo di pacchetti **TCP** con flag **RST**, normalmente questo tipo di pacchetti dovrebbero essere poco frequenti perché dovuti a errori di rete non malevoli, nel nostro caso evidenziamo molti pacchetti con flag **RST** inviati nell'arco di pochissimi millisecondi, il che ci fa sospettare non si tratti di un comportamento lecito, bensì:

-nel semplice caso di una scansione, **che pare essere quello più probabile**, le connessioni con flag **RST** appartengono a chiusure adoperate dalla macchina target probabilmente per un impostazione che chiude in automatico le sessioni non **'utilizzate'**

-potrebbe trattarsi di un attacco automatizzato che invia pacchetti **RST** in maniera continuativa, al fine di **distarre** eventuali sistemi di controllo e monitoraggio da attacchi più pericolosi.

questo perché di fatto il flag **RST** indica che vengono terminate le connessioni tra la macchina attaccante e quella target, il che istintivamente non ci fa pensare ad un attacco diretto, ma quanto più probabilmente appunto ad un tentativo di occultamento per altre operazioni potenzialmente **più dannose**.

-il traffico flaggato **RST** è bidirezionale, supponendo che l'attaccante sia ...200.150, in quanto è lui ad iniziare lo scan nella rete nel punto 1 della nostra analisi, questo potrebbe indicare che sono state aperte diverse connessioni poi chiuse per **inutilizzo**, come già detto, o per un **protocollo di sicurezza**,

o ancora potrebbe trattarsi di un escamotage da parte dell'attaccante, che simula chiusure al fine di non essere rilevato e "confondere le sue tracce" o eludere/rendere più complessi i controlli manuali sul traffico.

Procediamo quindi analizzando le connessioni chiuse dalla macchina target

3.1 Filtraggio RST macchina target

tcp.flags.reset == 1 && ip.src == 192.168.200.100									
No.	Time	Source	Destination	Protocol	Length	Info			
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
33	36.775619454	192.168.200.100	192.168.200.150	TCP	66	41384 → 23	[RST, ACK]	Seq=1	Ack=1 Win=64256 Len=0 TSval=810535439 TSecr=4294952466
34	36.77592497	192.168.200.100	192.168.200.150	TCP	66	50128 → 111	[RST, ACK]	Seq=1	Ack=1 Win=64256 Len=0 TSval=810535439 TSecr=4294952466
39	36.775961964	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	36.77600953	192.168.200.100	192.168.200.150	TCP	66	33062 → 89	[RST, ACK]	Seq=1	Ack=1 Win=64256 Len=0 TSval=810535439 TSecr=4294952466
86	36.777893286	192.168.200.100	192.168.200.150	TCP	66	33842 → 445	[RST, ACK]	Seq=1	Ack=1 Win=64256 Len=0 TSval=810535441 TSecr=4294952466
87	36.777912717	192.168.200.100	192.168.200.150	TCP	66	46990 → 139	[RST, ACK]	Seq=1	Ack=1 Win=64256 Len=0 TSval=810535441 TSecr=4294952466
88	36.777986759	192.168.200.100	192.168.200.150	TCP	66	60632 → 25	[RST, ACK]	Seq=1	Ack=1 Win=64256 Len=0 TSval=810535441 TSecr=4294952466
89	36.77831265	192.168.200.100	192.168.200.150	TCP	66	37282 → 53	[RST, ACK]	Seq=1	Ack=1 Win=64256 Len=0 TSval=810535441 TSecr=4294952466
170	36.781989537	192.168.200.100	192.168.200.150	TCP	66	45648 → 512	[RST, ACK]	Seq=1	Ack=1 Win=64256 Len=0 TSval=810535445 TSecr=4294952466
273	36.789881139	192.168.200.100	192.168.200.150	TCP	66	51396 → 514	[RST, ACK]	Seq=1	Ack=1 Win=64256 Len=0 TSval=810535453 TSecr=4294952467
1875	36.829275924	192.168.200.100	192.168.200.150	TCP	66	42048 → 513	[RST, ACK]	Seq=1	Ack=1 Win=64256 Len=0 TSval=810535493 TSecr=4294952471

filtro:

tcp.flags.reset == 1 && ip.src == 192.168.200.100

La macchina 192.168.200.100 invia pacchetti **TCP-RST** sulle seguenti porte ai fini di terminare le connessioni, confrontiamo ora con l'invio di richieste da parte dell'attaccante (192.168.200.150) flaggate **SYN-ACK**, al fine di verificare se ci siano effettivamente connessioni che risultano aperte, ma non vengono chiuse.

3.2 Filtraggio SYN macchina attaccante

No.	Time	Source	Destination	Protocol	Length	Info			
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=0
19	36.774685505	192.168.200.150	192.168.200.100	TCP	74	23 → 41384	[SYN, ACK]	Seq=0	Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM TSval=4294952466 TSecr=0
28	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 TSval=4294952466 TSecr=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=0
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=0
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=0
57	36.776984828	192.168.200.150	192.168.200.100	TCP	74	445 → 33842	[SYN, ACK]	Seq=0	Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM TSval=4294952466 TSecr=0
59	36.776984961	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 TSval=4294952466 TSecr=0
61	36.776985043	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 TSval=4294952466 TSecr=0
63	36.776985123	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 TSval=4294952466 TSecr=0
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 TSval=4294952466 TSecr=0
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 TSval=4294952467 TSecr=0
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 TSval=4294952471 TSecr=0

filtro:

tcp.flags.SYN == 1 && tcp.flags.ACK == 1 && ip.src == 192.168.200.150

Con un occhio attento possiamo notare dal confronto che la connessione aperta sulla porta 80

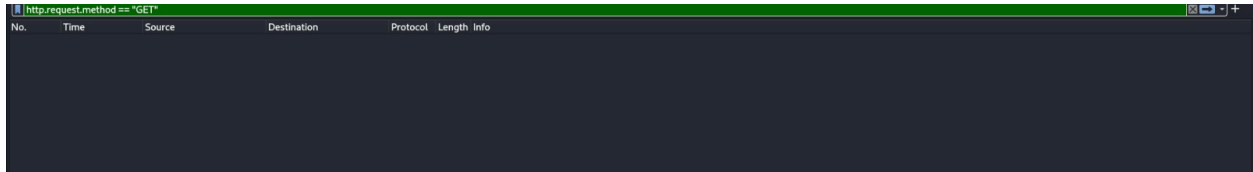
viene aperta (e terminata nell'immagine prima) per due volte, ed è l'unica a subire questo trattamento, segnalandoci che questo potrebbe essere stato effettivamente l'obiettivo dell'attaccante, ovvero sfruttare la porta 80, che genericamente è configurata per il traffico di rete

e di conseguenza accetta connessioni da praticamente chiunque, per poter lanciare attacchi come

SQL injection

al fine di sfruttare vulnerabilità note nella/e web app ospitate sulla macchina. Come ad esempio una vecchia versione di un Framework o di un editor di applicazioni web (es. Wordpress).

4 Verifica di attacchi già avvenuti o in corso



A screenshot of a Wireshark packet capture. The filter bar at the top shows 'http.request.method == GET'. The packet list below shows a single packet (No. 2) at time 23.764214995, from source 192.168.200.100 to destination 192.168.200.150, using TCP. The packet details pane shows the HTTP request structure.

No.	Time	Source	Destination	Protocol	Length	Info
2	23.764214995	192.168.200.100	192.168.200.150	TCP	74	80 → 53060 [SYN, Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=810522427 TSecr=4294952]

Per effettuare un controllo su eventuali attacchi già avvenuti abbiamo filtrato i diversi tipi di richieste (GET,POST...) comunemente utilizzati negli attacchi alle web app o ai server, non risulta esserci nessun risultato incriminante

(l'analisi è stata ripetuta con diversi metodi, nell'immagine è mostrato solo GET)

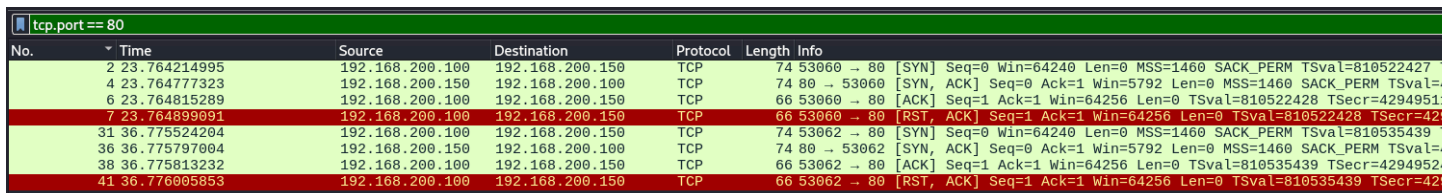
Questo ci porta a dedurre che l'attaccante potrebbe aver banalmente scansionato i servizi inerenti la porta 80, procediamo ad isolare le richieste inerenti la porta incriminata.

5 Esame porta 80

Andiamo ad esaminare con più attenzione il traffico sulla porta 80:

Filtro

tcp.port == 80



A screenshot of a Wireshark packet capture with the filter 'tcp.port == 80'. The packet list shows several packets (No. 2, 4, 6, 7, 31, 36, 38, 41) at various times, all from source 192.168.200.100 to destination 192.168.200.150, using TCP. The packet details pane shows the TCP and HTTP details for the selected packet (No. 2).

No.	Time	Source	Destination	Protocol	Length	Info
2	23.764214995	192.168.200.100	192.168.200.150	TCP	74	80 → 53060 [SYN, Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=810522427 TSecr=4294952]
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=810522428 TSecr=4294952
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=4294952
7	23.764899091	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=4294952
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=4294952
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=810535439 TSecr=4294952
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=4294952
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=4294952

Da questo filtro possiamo vedere con più chiarezza che la porta 80 effettua due **“Three Way HandShake”**, la soluzione più probabile in assenza di comandi inviati come visto nel punto 4 è che si tratti di una **scansione dei servizi inerenti la porta 80**.

5 Soluzione

Il nostro attaccante ha probabilmente **aperto** una connessione sulla porta, **scansionato** i servizi ai fini di trovare le **vulnerabilità** e ha poi chiuso la sessione.

Magari perché intende prepararsi per sfruttare le vulnerabilità trovate in futuro, dopo aver identificato il metodo di attacco adeguato e preparato il payload malevolo da iniettare.

abbiamo quindi identificato

- una scansione della rete per identificare il target

- una scansione delle porte sulla macchina target (nmap...?)

- un comportamento sospetto (possibile scansione dei servizi) inerenti la porta 80 (nmap,whatweb...)

5 Mitigazioni possibili e consigliate

Mitigazioni a lungo termine

Segmentazione della rete interna

Implementare VLAN e firewall per isolare segmenti critici della rete, limitando la propagazione di eventuali attacchi e riducendo la superficie di rischio.

Separare gli ambienti di rete inerenti a ogni settore può aiutare a prevenire accessi non autorizzati. Nel nostro caso infatti l'attaccante era all'interno della stessa rete del target.

Rinforzo del web server/web app

Disabilitare moduli e **servizi** non necessari, applicare costanti **aggiornamenti** di sicurezza inerenti ai **framework** utilizzati e configurare correttamente i permessi di accesso.

Questo riduce il rischio di **exploit** derivanti da vulnerabilità note.

Implementare Web Application Firewall (WAF)

Proteggere la porta 80 con un WAF c per bloccare tentativi di SQL Injection, LFI, XSS...

senza dubbio una difesa estremamente consigliata quando si tratta di proteggere web server/app.

Formazione del personale IT

Nel caso l'azienda disponga di un tecnico adibito alla reazione in caso di attacchi, possiamo aiutarlo a prevenire con maggior efficacia segnalando quanto emerge dal nostro report.

BONUS

Siete chiamati a progettare le difese di questo scenario:

Esercizio Bonus 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

Consegna:

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

Proponente al cliente due soluzioni,

- una economica (massimo 500 euro) e
- una più costosa (massimo 2500 euro) Eventuali altre specifiche richieste (non specificate) potete inventare

B.1 Valutazione delle vulnerabilità e punti di attacco

1.1 Attacchi alla VPN

- La VPN utilizzata per stabilire un canale sicuro di comunicazione tra cliente e macchina potrebbe essere soggetta ad attacchi come **Bruteforce** che tentano di indovinare le credenziali di accesso tramite l'uso di programmi o dizionari per poi stabilire una falsa connessione "sicura" tra la macchina attaccante e il target

1.2 Attacchi MITM

- Anche se il traffico è cifrato attraverso una VPN, un attaccante potrebbe intercettare e manipolare i dati che viaggiano tra il cliente e il macchinario. Ciò può avvenire se la configurazione della VPN è errata o se l'autenticazione non è forte.

Una cifratura debole delle comunicazioni potrebbe permettere al malintenzionato posizionarsi 'nel mezzo' (appunto MITM) tra macchina e cliente, dandogli accesso ai dati cifrati.

1.3 Vulnerabilità Windows 10/Windows Firewall

- Anche con un firewall attivo, gli attaccanti possono sfruttare vulnerabilità in diversi modi.

Abbiamo visto e testato ad esempio nel corso del nostro corso come sia possibile usare exploit noti come **eternal blue** per accedere al servizio **SMB** proprio sulla macchina Windows 10 e di come questo abbia permesso il diffondersi di ransomware come 'wannacry'.

La versione di windows 10, firewall o meno, va verificata a livello di aggiornamenti e integrità, prima di poterla considerare sufficientemente **sicura**.

1.4 Porte USB:

- Sebbene le pendrive siano disabilitate, le porte USB possono ancora essere utilizzate per altri tipi di dispositivi, come tastiere, mouse ecc..

Un attaccante potrebbe collegare una **tastiera USB** o un **mouse USB** a un sistema per eseguire attacchi **di tipo "BadUSB"**. In questi attacchi, dispositivi USB come tastiere o mouse sono modificati per agire da dispositivi di input malevoli, simulando comandi di sistema per compromettere il computer.

1.5 Software in C99:

- Essendo il software scritto in C99, potrebbero esserci vulnerabilità nel codice. Anche se non c'è scrittura sul sistema operativo, Il linguaggio C permette l'uso di array e buffer senza alcun controllo automatico sul loro dimensionamento, quindi se un buffer non è gestito correttamente, un attaccante potrebbe sfruttare questa vulnerabilità per eseguire codice malevolo o alterare il comportamento del programma.

questo tipo di attacco prende infatti il nome di **(Buffer Overflow)**

1.6 Propagazione di malware da macchine aziendali differenti

- Se un malware compromette un computer nella rete interna (es. un PC del cliente connesso alla VPN), potrebbe propagarsi alla macchina industriale e provocare, ad esempio, lo spegnimento del firewall o l'apertura di porte e backdoor.
- **1.7 Attacchi di ingegneria sociale o phishing**
 - Se un dipendente o un operatore del macchinario riceve una email di **phishing** con un allegato malevolo, potrebbe aprire un **eseguibile infetto** o inserire credenziali su un sito fake. Altrimenti, potrebbe essere un altro utente dell'azienda a ricevere un'email che permette la propagazione di un malware alla nostra macchina target.
- Questo tipo di attacco renderebbe del tutto superfluo l'uso di un Firewall o della VPN.

-

B.2 Proposte di sicurezza

Soluzione economica (<500 euro)

1. 1. Migliorare la sicurezza della VPN

- Implementazione di **autenticazione a due fattori (2FA)**.
- Limitazione dei tentativi di login falliti e blocco IP sospetti ai fini di prevenire eventuali bruteforce.

2. Configurazione avanzata del firewall

- Disabilitare porte e servizi non necessari.
- Implementare regole di accesso restrittive per limitare il traffico.

3. Controllo delle porte USB

- Disabilitare totalmente le porte USB tramite policy di gruppo (GPO).
- Implementare software gratuito per il monitoraggio delle connessioni USB.

4. Verifica delle vulnerabilità di Windows 10

- Installazione di aggiornamenti che permettano di eludere vulnerabilità critiche alla sicurezza come visto prima.
- Scansione periodica con Windows antivirus gratuito.

5. Sensibilizzazione e formazione del personale

-
- Sessioni di formazione base sulla sicurezza informatica e riconoscimento del phishing.

p.s. non conosciamo l'entità dell'azienda perciò questo costo non è correttamente stimabile

Costo stimato:

circa 400-500 euro (maggior parte dei costi associata alla configurazione e alla formazione del personale).

Soluzione avanzata (<2500 euro)

Obiettivo: implementare una protezione più completa con strumenti di monitoraggio avanzati.

1. Sicurezza avanzata della VPN

- Utilizzo di VPN con **certificati digitali e autenticazione multi fattore avanzata**.
- Analisi e monitoraggio dei log di accesso alla VPN tramite SW dedicati.

2. Firewall e segmentazione della rete

- Implementazione di un **firewall hardware dedicato** (es. pfSense)
- Creazione di una **VLAN** separata per isolare il macchinario (o i macchinari) dalla rete aziendale.

3. Monitoraggio del traffico

- Installazione di un **sistema IDS/IPS** open-source o a pagamento per rilevare traffico anomalo. Se la nostra macchina è un web server o ospita una web app, allora preferiremmo un **WAF** essendo più adatto alle nostre esigenze.

4. Protezione avanzata del software C99

-
- Revisione del codice per rilevare e mitigare vulnerabilità **buffer overflow** e gestione errata della memoria.

5. Sicurezza USB avanzata

- Acquisto di un dispositivo hardware di controllo USB come **USB Firewall** per bloccare dispositivi non autorizzati.

6. Soluzione di antivirus avanzato

- Installazione di una antivirus avanzato (**Microsoft Defender** ad esempio...)per monitorare attività sospette.

7. Simulazioni di phishing e formazione avanzata

- Implementazione di test periodici di phishing con report dettagliati.

p.s. per calcolare il costo della formazione di nuovo avremmo bisogno di stimare almeno le dimensioni dell'azienda, che non ci vengono fornite

Costo stimato:

2000-2500 euro destinati all'acquisto di hardware firewall, software di monitoraggio, revisione codice e formazione avanzata.

Daniele Balani