Report Esercizio 07/02/2025

Threat Intelligence & IOC Wireshark Leonardo Catalano

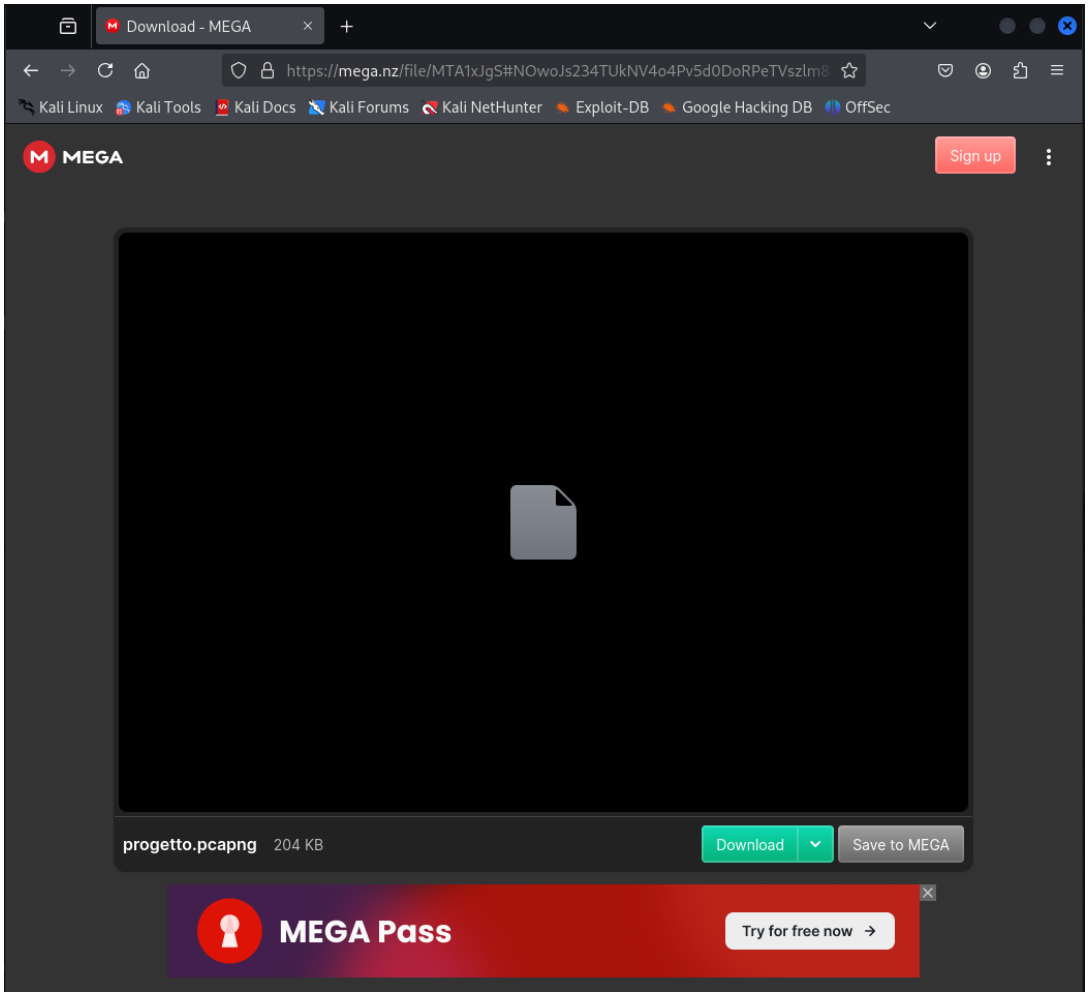
“La traccia di oggi ci chiede di effettuare un’analisi di una cattura di rete effettuata con Wireshark.”

Le fasi da effettuare saranno le seguenti:

1. Identificare ed analizzare eventuali IOC, (ovvero evidenze di Attacchi in corso).
   In base agli IOC trovati, descrivere eventuali potenziali vettori di attacco utilizzati.
2. Consigliare un’azione per ridurre gli impatti dell’attacco attuale.

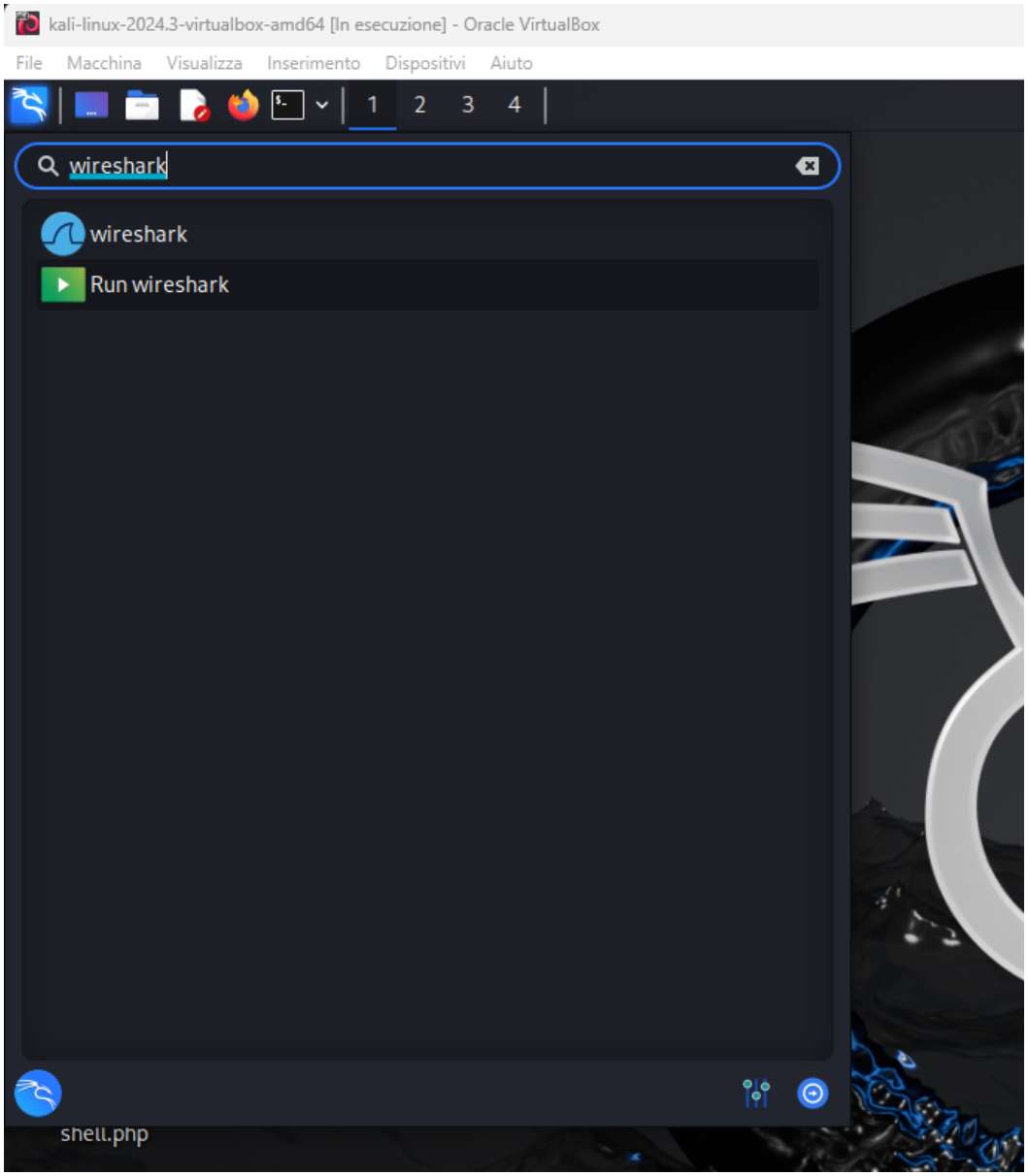
-Macchina Kali Linux:

Per scaricare il file da Kali bisogna lasciare la VM in scheda di rete NAT e dal browser andiamo a scaricare il file Wireshark della cattura della rete.
Per far ciò ho copia incollato il link del file fornito dalla traccia sul browser di Kali.

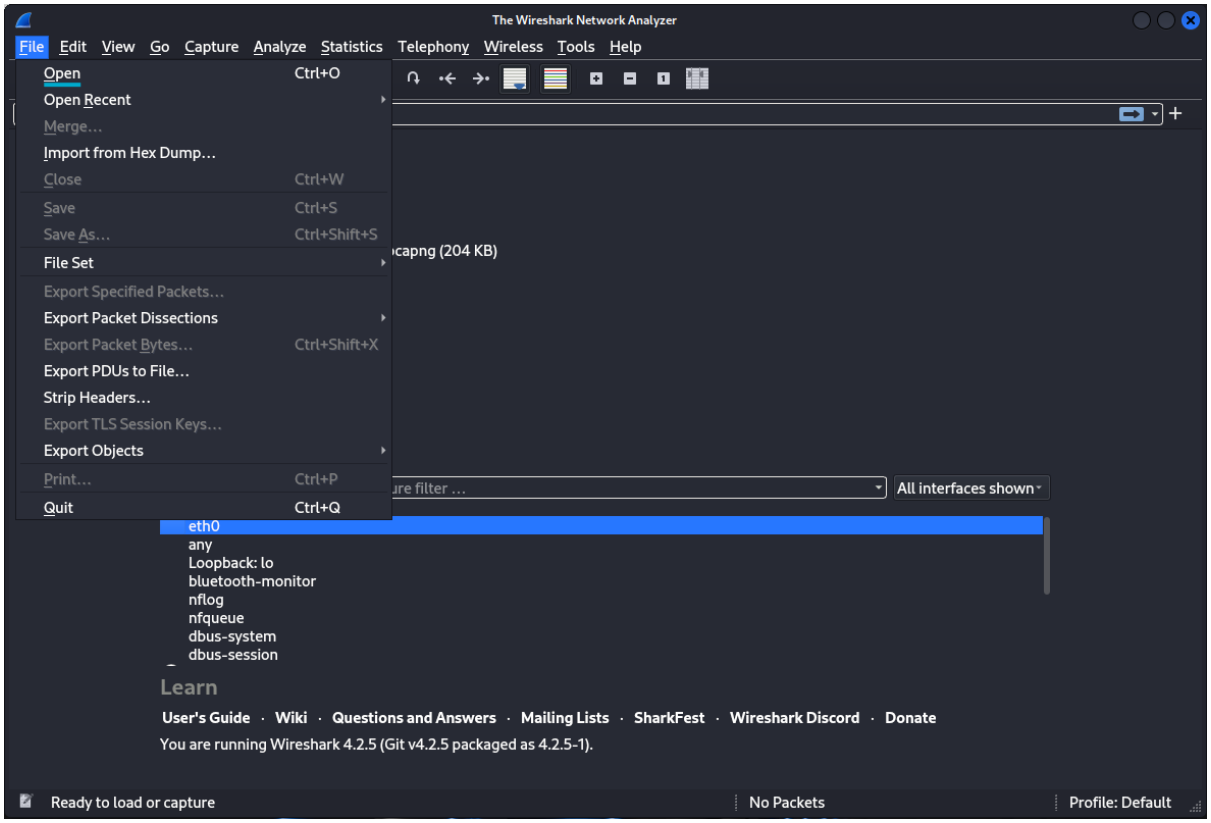

Download - MEGA
https://mega.nz/file/MTA1xJgS#NOwoJs234TUkNV4o4Pv5d0DoRPeTVszlm8
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
MEGA
Sign up
progetto.pcapng
204 KB
Download
Save to MEGA
MEGA Pass
Try for free now

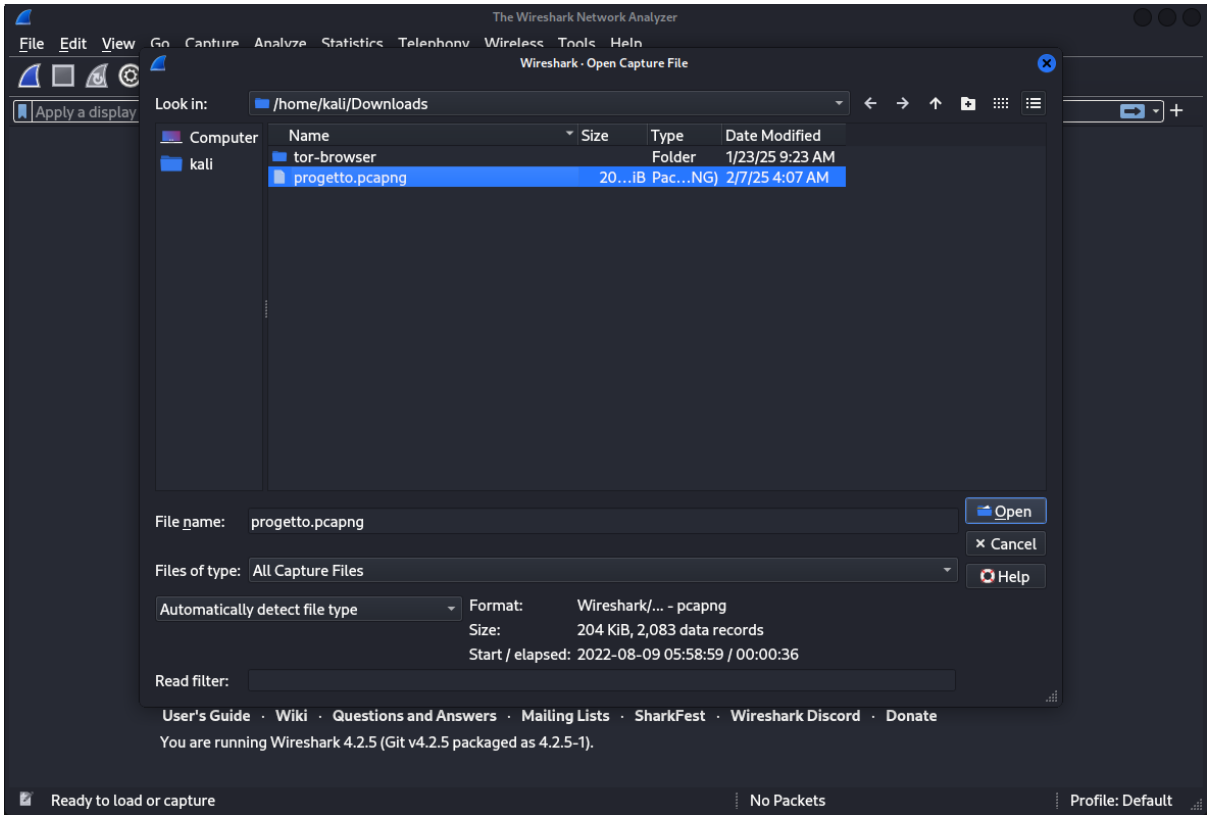
-Wireshark:

Per poter aprire la scansione effettuata cerchiamo il programma Wireshark dalla barra di ricerca di Kali:

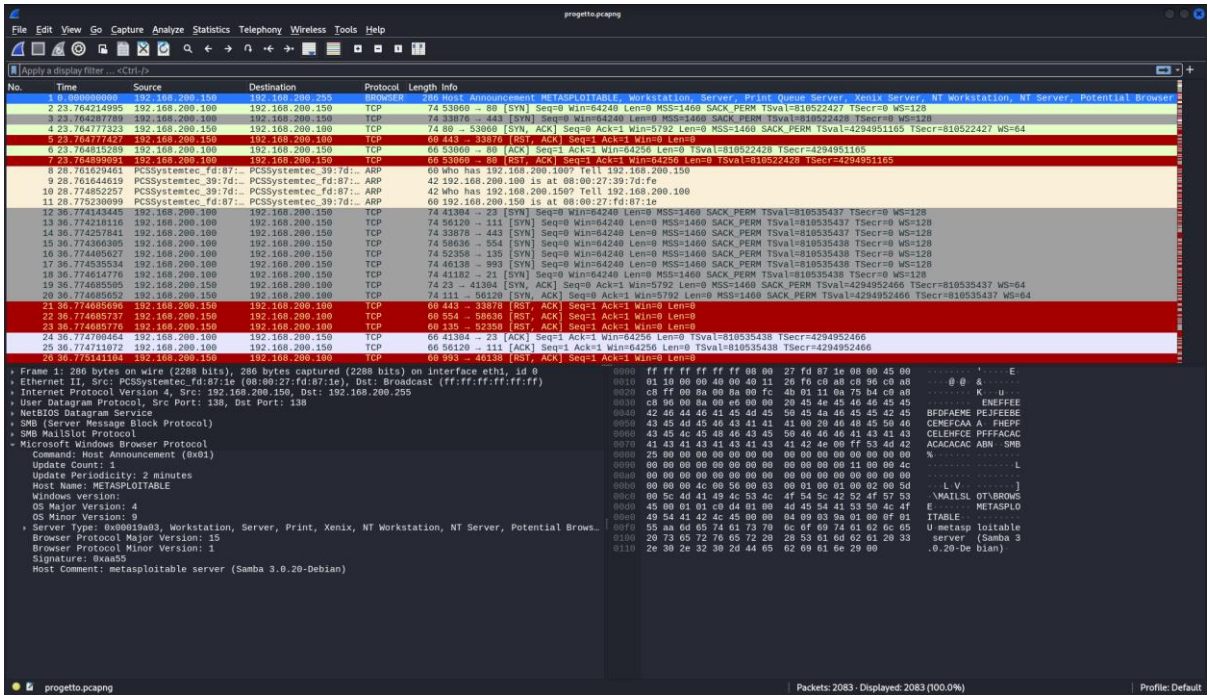


Una volta aperto Wireshark dal menu' File scegliamo l'opzione Open e andremo nella directory Download di Kali per aprire il file "progetto.pcapng" .



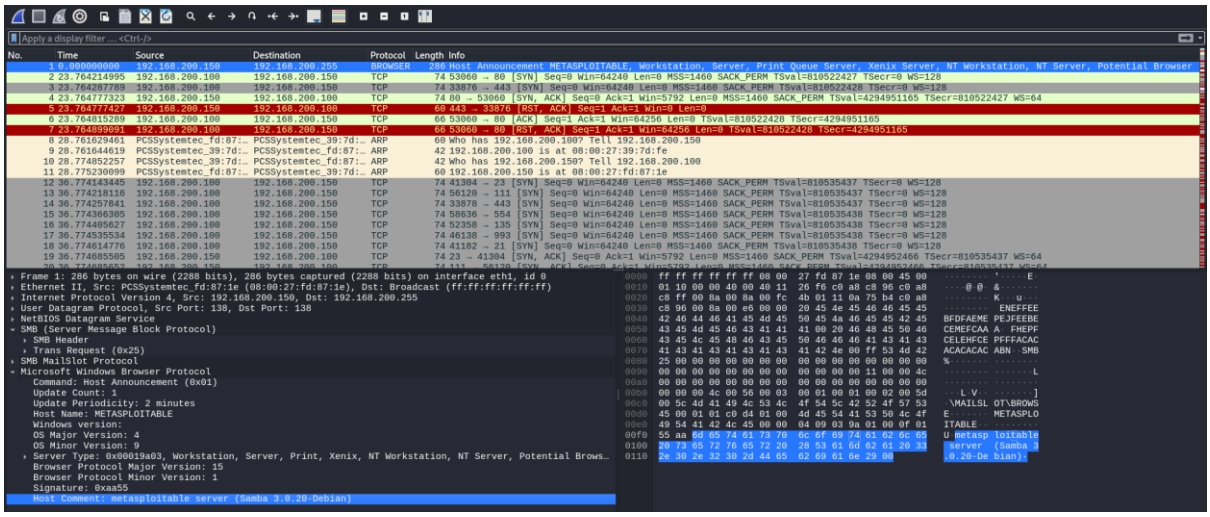


Selezionato il file avremo la seguente schermata con tutti i passaggi della scansione di Wireshark:



-Analisi della scansione con gli Ioc (Indicatori di Compromissione):

Ora andrò ad analizzare i vari passaggi passo passo:



Dalla scansione fornita si evidenziano subito 2 host, che si comunicano e che fanno parte della stessa rete:

1)Host --> 192.168.200.150 (per specificare quest'host userò solo .150)

2)Host --> 192.168.200.100 (per specificare quest'host userò solo .100)

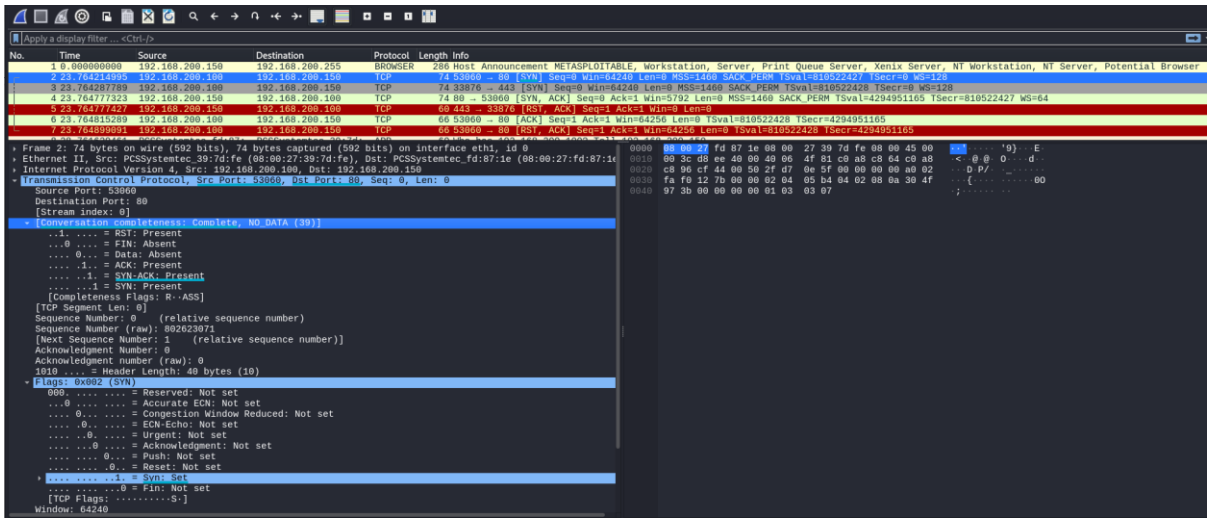
Come 1* log di pacchetto abbiamo una chiamata dall'host .150 al'indirizzo di Broadcast .255, per farsi conoscere agli altri host nella rete.

Lo vediamo nella sezione delle Info "Host Announcement Metasploitable, Workstation, Server, Print Queue Server, NT Workstation).

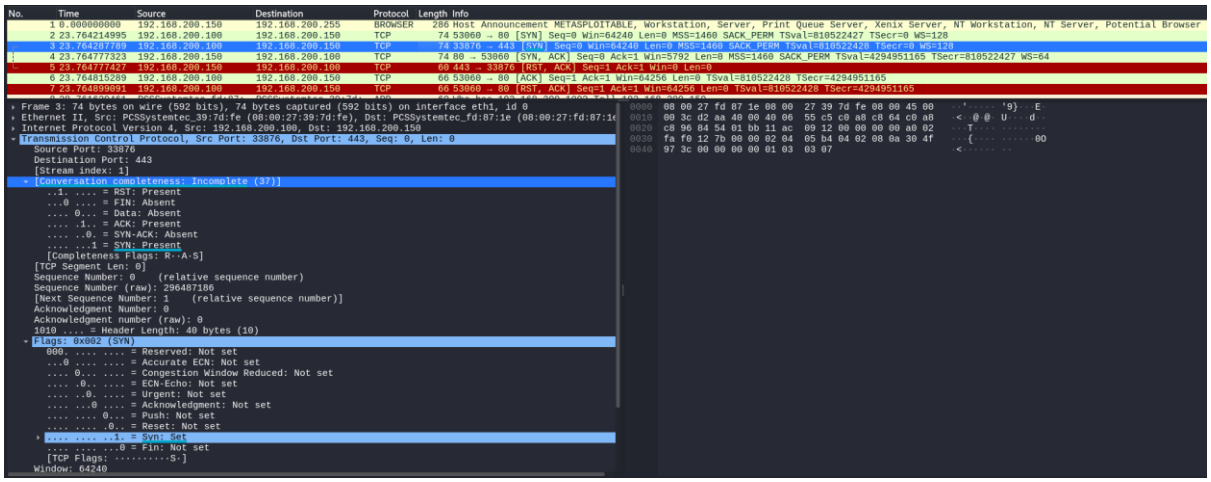
Inoltre nei dettagli nella sezione in basso (come da screen) "Microsoft Windows Browser Protocol", sottosezione "Server Type" vediamo L'Host Comment: metasploitable server (Samba 3.0.20-Debian), ciò sta a significare che questo indirizzo

specifico è dedicato ad un server Metasploitable “Samba 3.0.20 -Debian” versione vecchia del 2005.

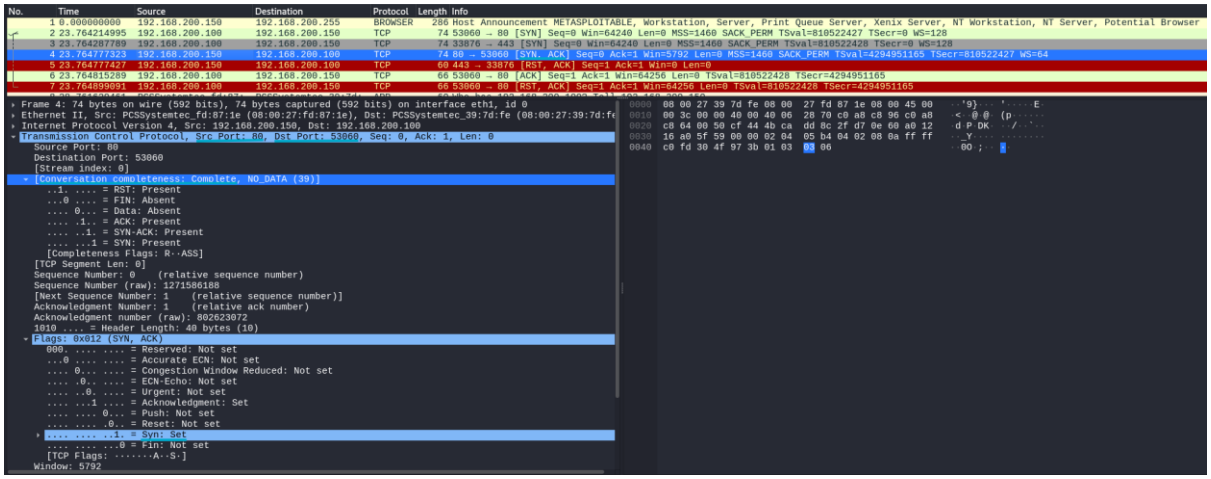
Passando al 2* log possiamo notare che il 2* host .100 manda un pacchetto TCP SYN al .150 dalla porta 53050 verso la porta 80 del .100 .



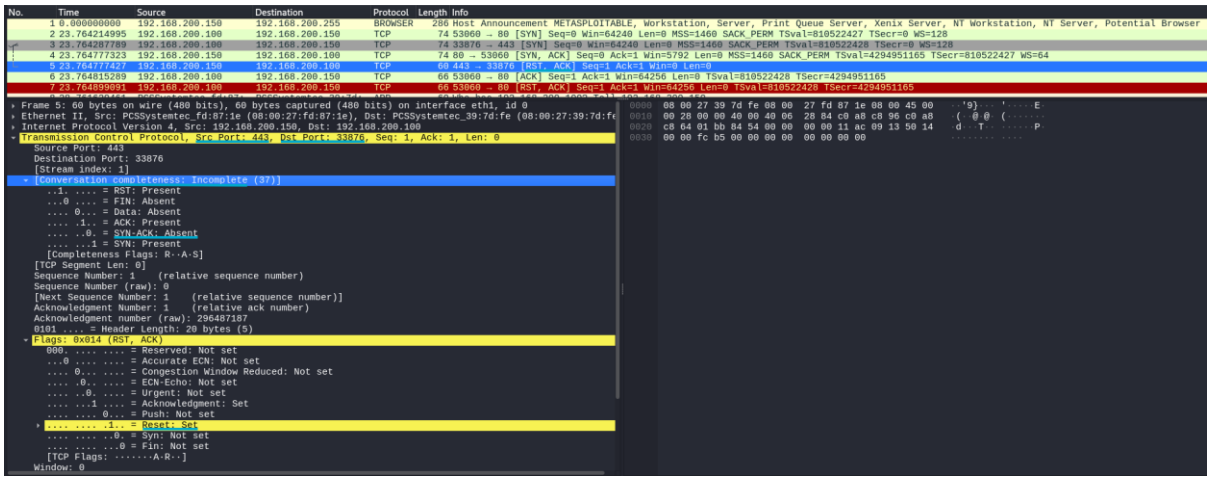
Nel 3* log manda un altro pacchetto TCP SYN al .150 ma stavolta dalla porta 33876 verso la porta 443 del .100 .



Nel 4* log il .150 invia una risposta al .100 restituendo un SYN, ACK ciò vuol dire che la comunicazione è andata a buon fine dalla sua porta 80 alla porta 53060 del .100 che aveva precedentemente effettuato la richiesta.

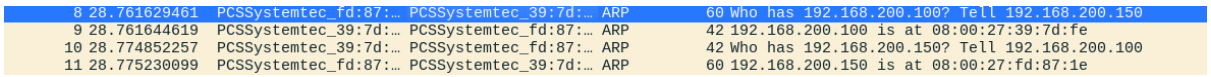


Nel 5* log invece il .150 invia una risposta al .100 restituendo un RST, ACK (di reset) ciò vuol dire che la richiesta non è andata a buon fine (quindi è probabile che la porta 443 sia chiusa), dalla sua porta 443 alla 33876.100 .

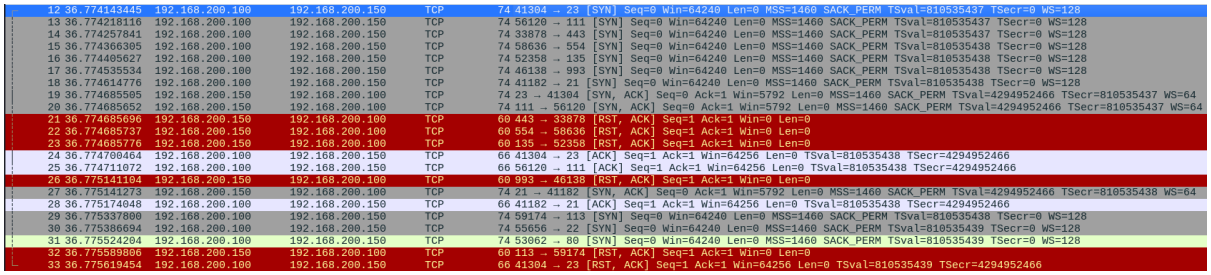


Prendendo questa **sequenza** come **base** nei futuri log di pacchetti potremmo trovare un **analogia/corrispondenza**, soltanto che sarà anche maggiore a livello di richieste e risposte, e li ci saranno degli scenari specifici di ciò che può accadere.

Dall'8* al 11* log avviene uno scambio di informazioni con gli indirizzi MAC tra i 2 host attraverso il protocollo ARP.



Da qui in poi (dal 12* log in poi) cominciano ad esserci delle anomalie e un traffico sospetto da parte della macchina .100 verso la macchina .150 , inviando più richieste nell'arco di pochi millisecondi di differenza (dallo screen siamo sempre a 36 secondi dall'inizio della scansione), da ciò si ripete lo schema sequenziale (**sequenza base**) che ho descritto prima come base.



12	36.774143445	192.168.200.100	192.168.200.150	TCP	74 41304 → 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 56120 → 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	36.774366305	192.168.200.100	192.168.200.150	TCP	74 58636 → 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
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 TSval=4294952466 TSecr=810535437 WS=64
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 TSval=4294952466 TSecr=810535437 WS=64
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	36.775337880	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

-Considerazioni:

Basandomi su ciò che ho analizzato e dai vecchi test con Wireshark per i vecchi esercizi (dos, metasploit “msfconsole”, reverse payload, nmap), in questo caso ho trovato un’analogia con l’Nmap, ma non è un Nmap normale è particolare, dove ogni volta si vanno a fare delle richieste con porte diverse a determinate porte (o servizi), quindi il .100 invia tante richieste al .150 in pochissimo tempo, che esso con i suoi tempi risponde.
Utilizza porte diverse come se volesse fare in modo di non farsi bloccare le richieste in una determinata porta.
(Un Nmap standard da una porta sola invia le richieste alle altre porte de target per vedere se siano aperte o meno).

Analizzando ho trovato una **sequenza di scenari** che si ripete fino alla fine della scansione.

Quello che può succedere è:

1) il .100 spamma le richieste e vede che il .150 non gli risponde a tutte, quindi glie le rimanda (c’è un time to live dei pacchetti).
2) il .100 vede anche dopo che glieli ha rimandati (quindi gli ha rimandato la richiesta) che non gli risponde e manda il pacchetto RST ACK (di reset) per stoppare la richiesta.
3) il.100 vede che il .150 non gli risponde e decide di non rimandare la richiesta e manda subito il pacchetto RST ACK (di reset) per stoppare la richiesta.
4) se la porta nel .150 è aperta, il .150 gli risponde con un SYN (e nei dettagli del log appare Conversation Complete).
5) se la porta nel .150 è chiusa, il .150 gli risponde con un RST ACK (di reset) per stoppare la richiesta (l’ho testato prima con un nmap singolo ad una porta su metasploit ed è cosi).

-Mitigazioni e Conclusioni:

Per prevenire queste situazioni, in cui si riscontra un traffico anomalo proveniente dalla stessa fonte (stesso indirizzo IP), si dovrebbe configurare (o installare se non c’è proprio) un Firewall in modo tale da bloccare tantissime richieste fatte in pochissimo tempo da un medesimo indirizzo IP.

Un’altro “problema” potrebbe essere che attraverso l’Nmap si scoprono quali sono le porte aperte (e i servizi associati ad essa), quindi se una porta non si usa, sarebbe buona prassi chiuderla.
Se la si deve lasciare aperta, si dovrebbe inserire una password per accedere poi a tale servizio.