S9L3,

BENVENUTI LUIGI

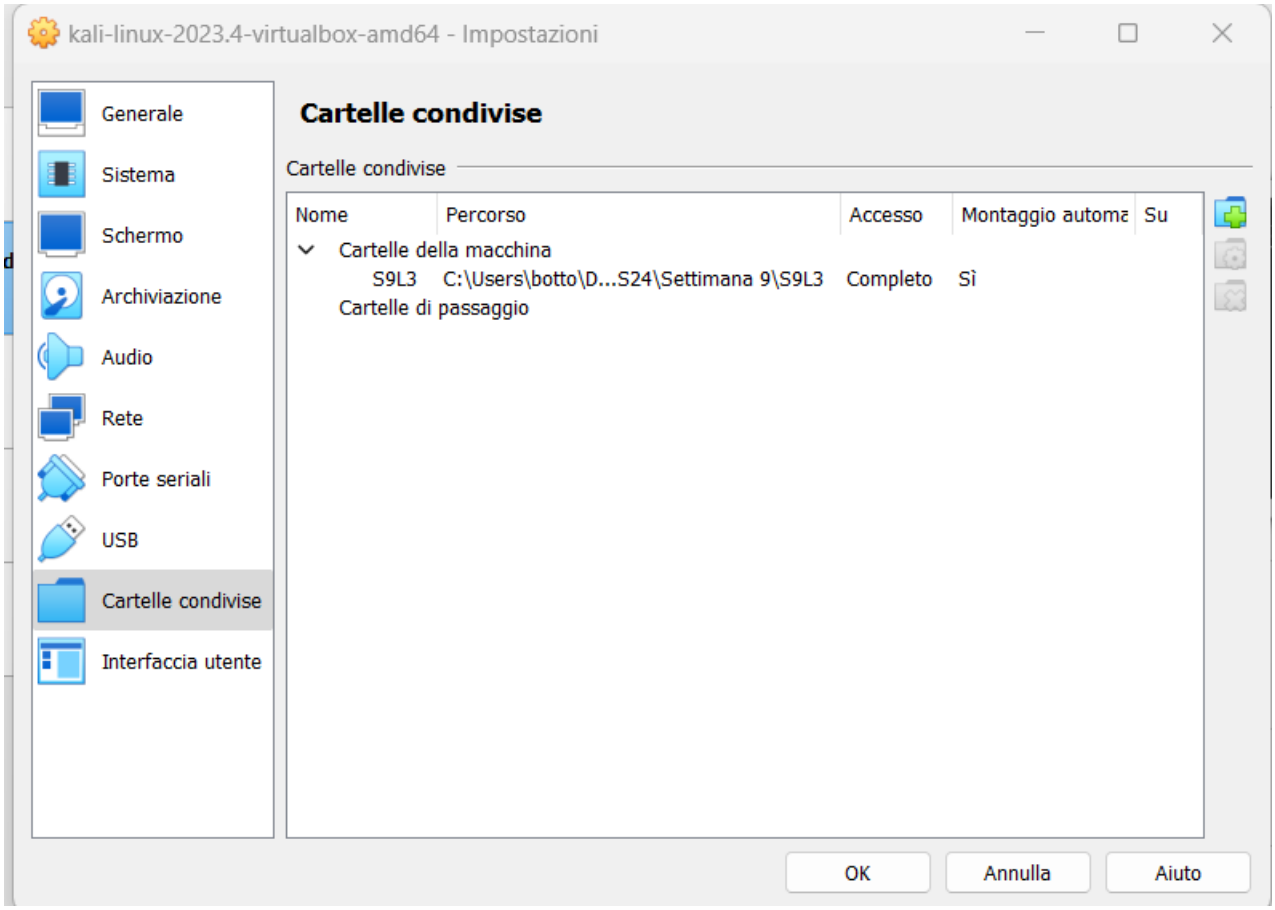
Traccia:

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.

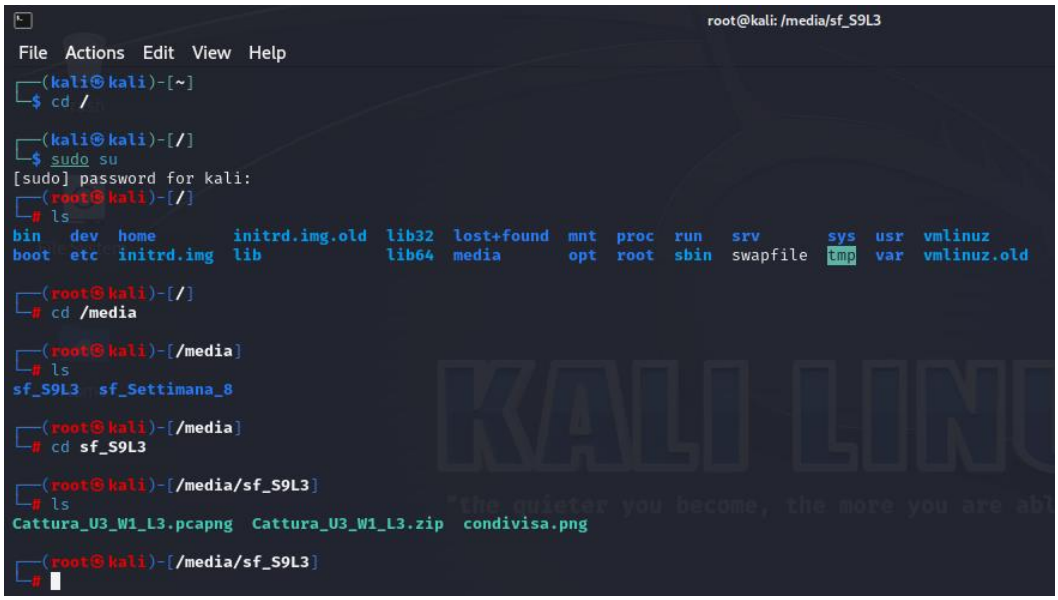
Per l'esercizio pratico di oggi, trovate in allegato una cattura di rete effettuata con Wireshark. Analizzate la cattura attentamente e rispondere ai seguenti quesiti: Identificare eventuali IOC, ovvero evidenze di attacchi in corso in base agli IOC trovati, fate delle ipotesi sui potenziali vettori di attacco utilizzati.

Consigliate un'azione per ridurre gli impatti dell'attacco.

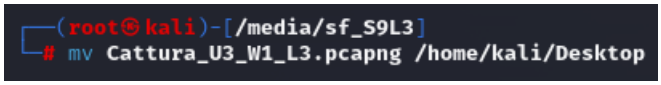
Condividiamo il file scaricato in una cartella condivisa con la macchina virtuale Kali Linux in Virtual Box.



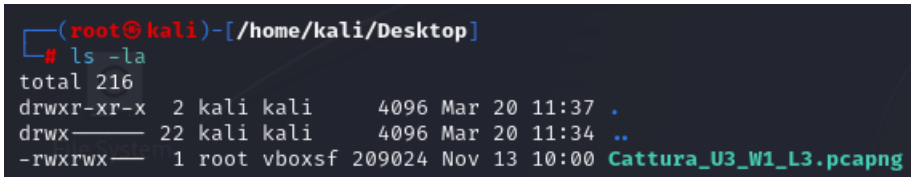
Individuiamo il path del file dal terminale.



E spostiamo il file sul desktop.



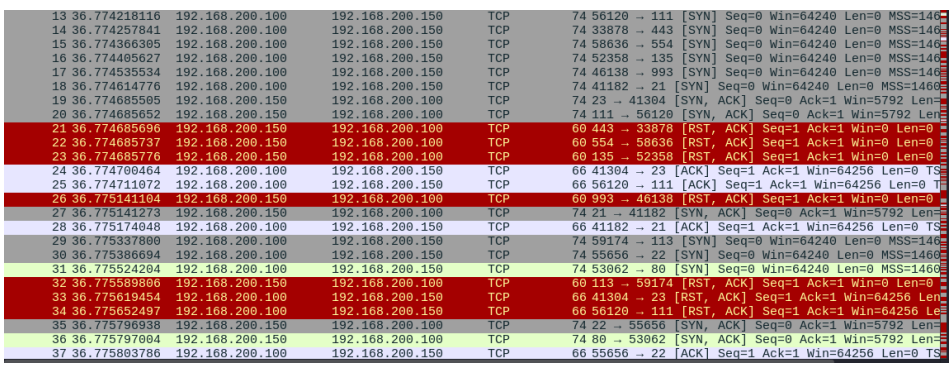
E verifichiamo quali permessi abbiamo sul file.



A questo punto vediamo quali aspetti prendere in considerazione nel monitoraggio:

1. Identificare eventuali IOC, ovvero evidenze di attacchi in corso;
2. In base agli IOC trovati, fate delle ipotesi sui potenziali vettori di attacco utilizzati;
3. Consigliate un'azione per ridurre gli impatti dell'attacco;
4. Potremmo configurare delle policy firewall per bloccare accesso a tutte le porta da parte di quel determinato attaccante, in modo tale da evitare che informazioni circa porta / servizi in ascolto finiscano nelle mani dell'attaccante.

Dalla cattura notiamo che ci sono un numero elevato di richieste TCP (SYN) su porte sempre diverse in destinazione.



Questo ci fa pensare ad una potenziale scansione in corso da parte dell'host 192.168.200.100 verso l'host target 192.168.200.150.

Questa ipotesi è supportata dal fatto che per alcune righe della cattura vediamo risposte positive del target [SYN+ACK] ad indicare che la porta è aperta; per altre, invece, notiamo la risposta [RST+ACK] ad indicare che la porta è chiusa.

Lato target, si potrebbero configurare delle regole firewall per respingere le richieste in entrata dall'host 192.168.200.100.

```
┌──(kali㉿kali)-[~]
└─$ sudo iptables -A INPUT -s 192.168.200.0/24 -j DROP
```

La nuova regola è stata inserita.

```
┌──(kali㉿kali)-[~]
└─$ sudo iptables -S
-P INPUT ACCEPT
-P FORWARD ACCEPT
-P OUTPUT ACCEPT
-A INPUT -s 192.168.200.0/24 -j DROP
```