Python on Kali Linux: A beginner’s guide to get started

infosecscout.com/python-kali-linux

Patrick Fromaget

Python is now the most used programming language in the world, with usage ranging from artificial intelligence to web development or even smartphone applications. For hackers and pen-testers, Python can also become a great ally on Kali Linux, to save time on many processes. Let’s see how to install and use Python on your favorite distribution.

On Kali Linux, a recent version of Python is already installed with dozens of libraries ready to use. No code editor is present, but most of them are available in the default repository and can be installed easily with APT.

Let’s start by checking your system installation, making sure Python is ready to use, and I’ll show you a few things you can try with Python on Kali Linux.

Getting Started with Python on Kali Linux

Even if coding in Python is maybe not the main goal for the typical Kali Linux user, everything is already set up for you on a fresh installation. Except choosing a text editor, there is almost nothing to do.

How to install Python on Kali Linux

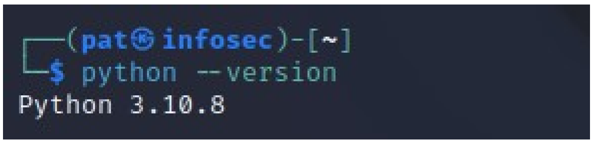
Python is already installed by default on Kali Linux. I kept the default options during the installation (including the essentials tools), and it was already there.

You can use this command to make sure Python is installed, and see which version:

```
python --version
```

It should be Python 3, maybe not the latest version, but one that is pretty recent. You can always uninstall it and get the latest version manually, but in most cases it's not worth the effort. Just do regular updates with APT, and it will get updated when ready:

```
sudo apt update
sudo apt upgrade
```



```
┌──(pat㉿infosec)-[~]
└─$ python --version
Python 3.10.8
```

By the way, if for any reason Python is not present on your system, you can install it with APT too:

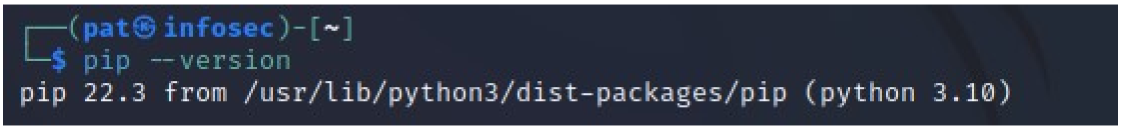
```
sudo apt install python
```

How to install PIP on Kali Linux

PIP is a package manager for Python. It's connected to a large database of libraries available online (Pyhton Package Index, or PyPi). It's already installed by default on Kali Linux, so you can use it right away.

You can use this command to make sure it's available and see the version:

```
pip --version
```



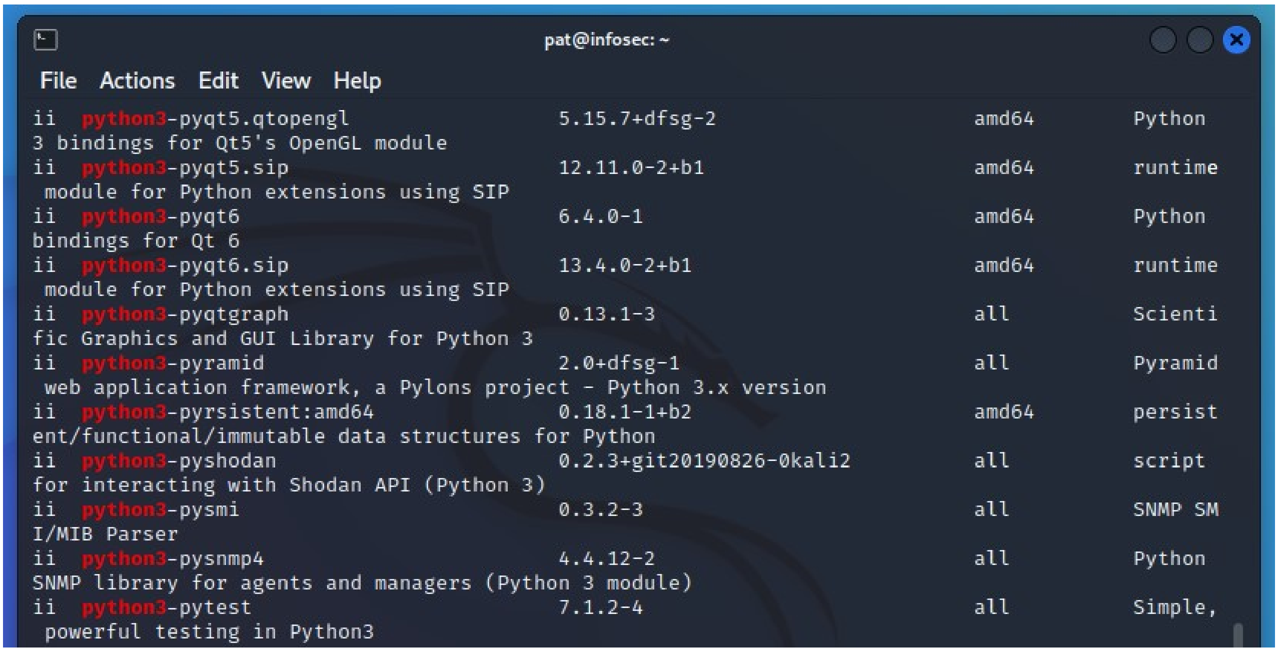
```
┌──(pat㉿infosec)-[~]
└─$ pip --version
pip 22.3 from /usr/lib/python3/dist-packages/pip (python 3.10)
```

I'll elaborate on libraries in the next paragraph, but basically, if you need any random library that is not available with APT (or pre-installed on Kali Linux), you can use this tool to download and install them quickly.

Python libraries on Kali Linux

Python is nothing without its libraries. Each library adds new functions that you can use in your Python scripts. You don't have to code everything yourself, you can rely on projects created by other developers, and just reuse their work in yours.

Dozens of libraries are pre-installed on a fresh Kali Linux system (probably more than 100 in fact). So, for most Python projects, you'll already have the libraries you need available.
Here is a screenshot of some of them:



```
ii  python3-pyqt5.qtopengl              5.15.7+dfsg-2                  amd64        Python
3 bindings for Qt5's OpenGL module
ii  python3-pyqt5.sip                   12.11.0-2+b1                   amd64        runtime
 module for Python extensions using SIP
ii  python3-pyqt6                       6.4.0-1                        amd64        Python
bindings for Qt 6
ii  python3-pyqt6.sip                   13.4.0-2+b1                    amd64        runtime
 module for Python extensions using SIP
ii  python3-pyqtgraph                   0.13.1-3                       all          Scienti
fic Graphics and GUI Library for Python 3
ii  python3-pyramid                     2.0+dfsg-1                     all          Pyramid
 web application framework, a Pylons project - Python 3.x version
ii  python3-pyrsistent:amd64            0.18.1-1+b2                    amd64        persist
ent/functional/immutable data structures for Python
ii  python3-pyshodan                    0.2.3+git20190826-0kali2       all          script
for interacting with Shodan API (Python 3)
ii  python3-pysmi                       0.3.2-3                        all          SNMP SM
I/MIB Parser
ii  python3-pysnmp4                     4.4.12-2                       all          Python
SNMP library for agents and managers (Python 3 module)
ii  python3-pytest                      7.1.2-4                        all          Simple,
 powerful testing in Python3
```

I got this list by using the "dpkg" command:
`sudo dpkg -l | grep python3`

I'll take and example later, but if you need a library that is not pre-installed, you can generally install it with APT, by using this syntax:
`sudo apt install python3-<library>`
Or by using the PIP tool I introduce earlier, like this:
`pip install <library>`

We're almost done, the only thing missing might be a fancy code editor, if you don't want to use the default notepad or a text editor in a terminal.

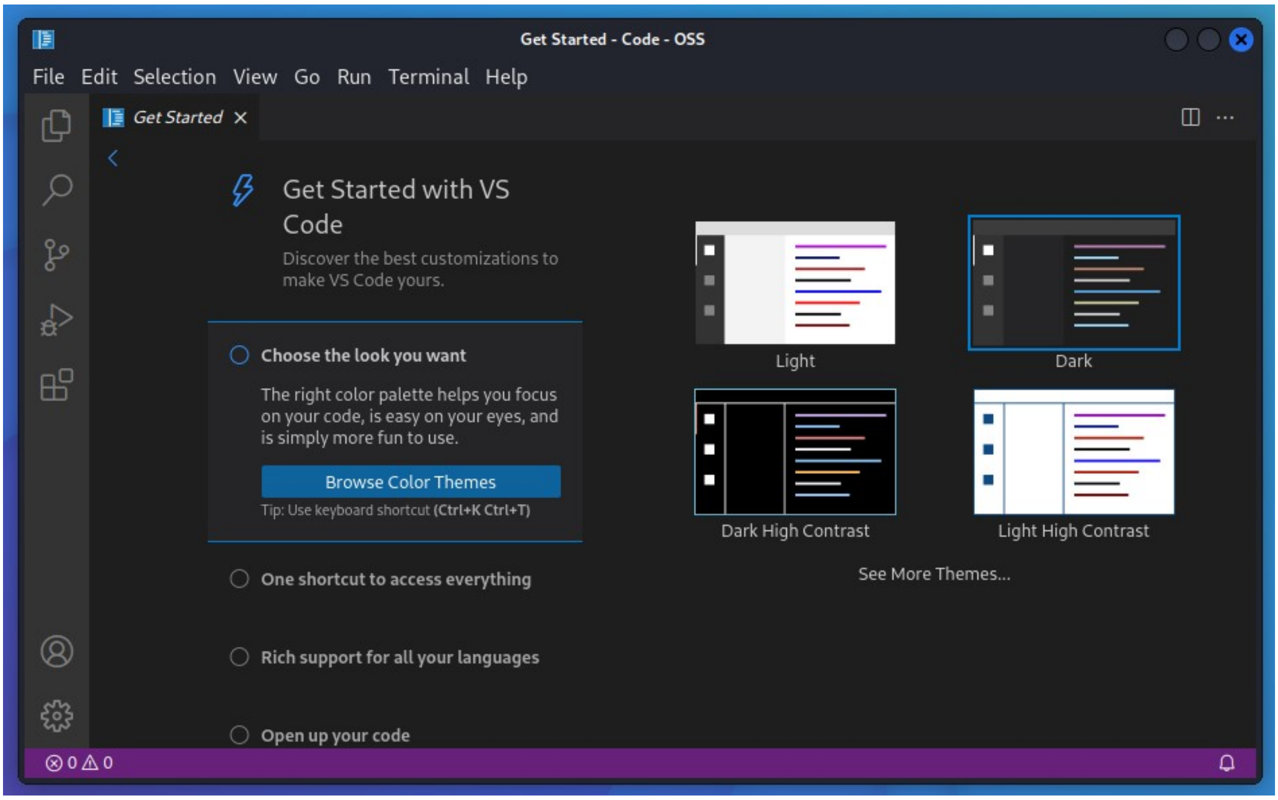
Python code editors you can use on Kali Linux

Except the default notepad or Nano, Kali Linux doesn't include any graphic text editor. However, Kali Linux is based on Debian, so most of them can be installed manually from the default repository, or by using packages provided by the developers.

That's a bit strange that they didn't include anything, when we see that everything else is taken care of. But I guess they weren't able to decide which one to choose :-).

Anyway, I tried to install VS Code and Geany, and it was done easily.
VS Code is not open-source, but the open-source version (code-oss) is available in the default repository. If you are used to Visual Studio Code, it should be good enough. You can install it with APT:

```
sudo apt install code-oss
```



Get Started with VS Code

Discover the best customizations to make VS Code yours.

○ Choose the look you want

The right color palette helps you focus on your code, is easy on your eyes, and is simply more fun to use.

Tip: Use keyboard shortcut **(Ctrl+K Ctrl+T)**

○ One shortcut to access everything

○ Rich support for all your languages

○ Open up your code

Same thing for Geany, that is available with:

```
sudo apt install geany
```

For other editors, you'll find either find repositories that you can add, Debian packages or source files to compile and install manually. For example, if you prefer Sublime Text, you can download the Debian package here (a repository is also available).

Nano is available directly by default, several options are available in the default repository, and plenty of editors have a Linux version that works on Kali. So make your choice and move on once you're ready.

Practical applications with Python on Kali Linux

All of this is very interesting, but Kali Linux is not really the first choice for Python developers. So, why should you use Python, and what can you do with it? Let's see a few real-life examples.

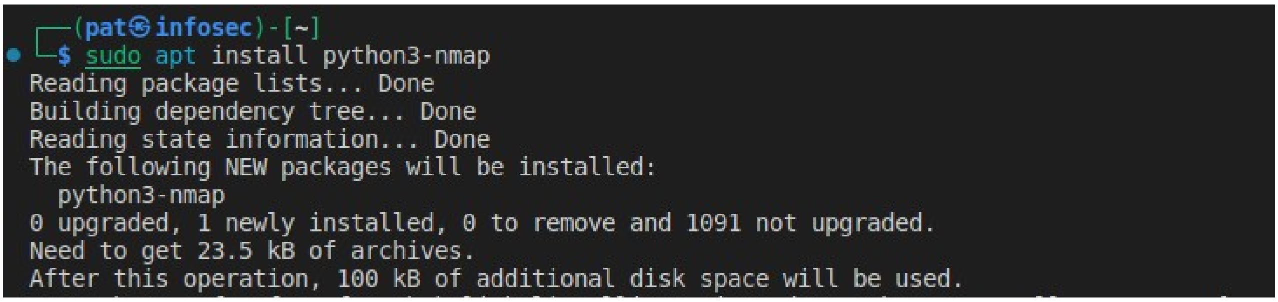
Scan network and run attacks based on the results

Kali Linux comes with many tools, that are generally used in command lines. So, basically, you rely on tools, but you do everything manually, in a terminal (or via the graphic interface is there is one).

For example, **Nmap is a popular tool to scan hosts on a network, but do you know that there is a library (python3-nmap) available to do it inside your Python scripts?**

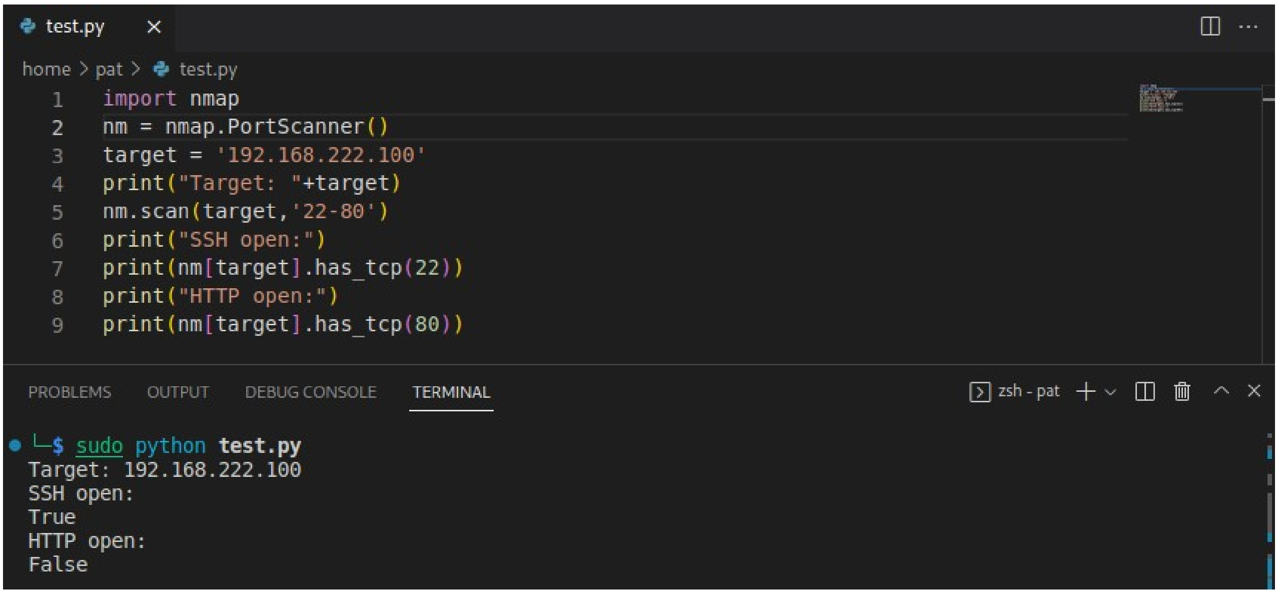
This way you can scan the network, and run actions based on the results you get. Maybe it's a deeper scan on specific devices, or adding the IP addresses in the queue list for other scripts or tools you have.

Strangely, the Nmap library is not installed by default, but you can install it with:
`sudo apt install python3-nmap`



```
┌──(pat㉿infosec)-[~]
└─$ sudo apt install python3-nmap
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  python3-nmap
0 upgraded, 1 newly installed, 0 to remove and 1091 not upgraded.
Need to get 23.5 kB of archives.
After this operation, 100 kB of additional disk space will be used.
```

You can then use it in your scripts to detect open ports on the network.
Here is a basic example:



```python
import nmap
nm = nmap.PortScanner()
target = '192.168.222.100'
print("Target: "+target)
nm.scan(target,'22-80')
print("SSH open:")
print(nm[target].has_tcp(22))
print("HTTP open:")
print(nm[target].has_tcp(80))
```

```
└─$ sudo python test.py
Target: 192.168.222.100
SSH open:
True
HTTP open:
False
```

This script will scan the ports open on my target (192.168.222.100), for a specific range (between 22 and 80). I then use the has_tcp() function to display a message to tell if SSH or HTTP are responding.

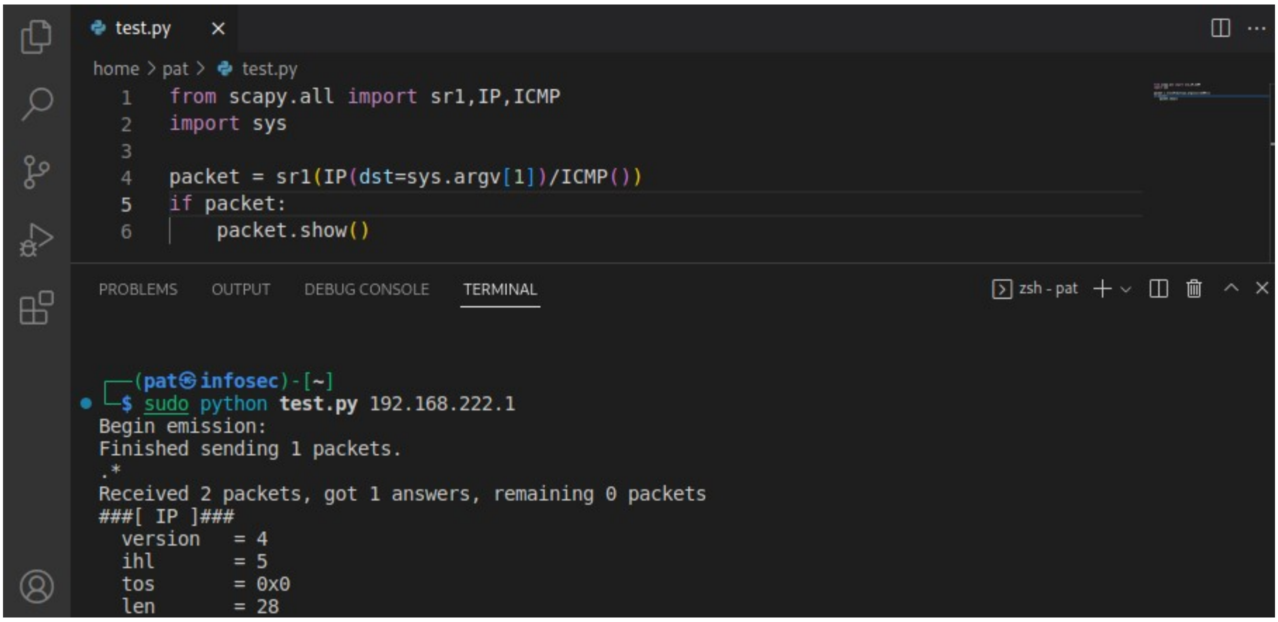
In a more concrete example, I could scan all the devices on the local network, and do something specific when I find one with the SSH port open (a brute force attack or something like that).

Note: If you use the version available on PyPi, the syntax will be slightly different, so make sure to follow the documentation corresponding to the library you use.

Packets sniffing and manipulation

Scapy is another tool you can use on Kali Linux to interact with network packets. It comes with a Python library, so you can do everything directly within your Python scripts.
Scapy and the corresponding Python library are installed by default on Kali Linux.

You can for example use the sniff() function to analyze packets on your network, and do specific actions depending on their content or destination. Or you can simply send a "ping" to any IP address, and do something if there is a response (like the port scanner code with Nmap).



```python
from scapy.all import sr1,IP,ICMP
import sys

packet = sr1(IP(dst=sys.argv[1])/ICMP())
if packet:
    packet.show()
```

```
┌──(pat㉿infosec)-[~]
└─$ sudo python test.py 192.168.222.1
Begin emission:
Finished sending 1 packets.
.*
Received 2 packets, got 1 answers, remaining 0 packets
###[ IP ]###
  version   = 4
  ihl       = 5
  tos       = 0x0
  len       = 28
```

You get the idea. It's the same thing as with Nmap. We take a tool that is available on Kali Linux, but use it in Python, so we can use loops, conditions and run different actions based on the results. All the good stuff that programmers like.

If you want to give it a try, pick the library corresponding to a tool you know and read the online documentation for more details about the functions available.

Coding malware

You may not know it, but most malware is written in Python. So whether you're a pen-tester or a budding hacker, you can use Python not only to make your life easier, but also to send attacks directly to a target and see how users react to the threat.

I do not endorse this type of practice, so I won't give you more details about this, but it's not a coincidence if Python is pre-installed on Kali Linux. Both types of users can benefit from it.

Building tools for you

Overall, Python is a good way to create custom tools or scripts for pen testing, auditing, etc.

Learning Python shouldn't be that complicated, whether you already have some experience with other languages or not. In most cases, you can rely on existing libraries, and follow the examples available in their documentation to quickly get results.

But if you are new to this, or want to be more confident with Python in general, you may need some help.
There are a few cheap courses available online that can help you reach this goal faster, here are a few ones I like:

- **The complete Ethical Hacking course (Python & Kali Linux)**
  Perfect for beginners in Python, as most of the course is about Python.
  It merges with ethical hacking near the end, but it's mostly a Python course.
- **Ethical Hacking Bootcamp: Zero to mastery**
  The 5 phases of ethical hacking, with Python practice for each of them (port scanner, password cracking, etc.)
- **Applied Ethical Hacking**
  It's like the top-level of ethical hacking course, once you already know the basics. You'll use most of the Kali Linux tools, Python, but also discover broader concepts like social engineering.

I gave you the basics to get started, now it's up to you to apply them in your projects and environment. Following a complete course can save you a lot of time, but it's not mandatory (you can always learn by yourself, it just requires more effort in general).