Security Testing With Python Scripts

Frida ZISKO
Department of Information Business, Mediterranean University of Albania, Tirana, Albania
E-mail address: frida.zisko@umsh.edu.al

Alma HYRA
Department of Information Technology, Mediterranean University of Albania, Tirana, Albania
E-mail address: alma.hyra@umsh.edu.al

Abstract
Cyber security is one of the main issues that is discussed today everywhere in the world. The development of technology has grown rapidly and it is important that this development should be done simultaneously with the increase in security. When we talk about the network, we must also consider its "Security". On the other hand, Python is a language which, especially in recent times, has received a great development and now we can say that it enjoys such a wide community. Our goal in this article is the advantages that this programming language offers in cases of cyber security issues. It has everything that cyber security professionals need to protect against cyber vulnerabilities and threats. It allows developers to do anything that relates to cyber security by detecting threats to system fixes. Python is a high-level, general-purpose, interpreted programming language for analyzing small networks. So, the implications of the study are for cybersecurity professionals and developers. They use the NetworkX grid tools imported into python. NetworkX provides many functions and generation facilities to read and write graphics in many formats. By analyzing these network nodes graphs, are identified the nodes that can be "bad". With Python scripts, you can scan ports and compromised device. In order to achieve this, we have obtained two concrete dataset which were obtained respectively from a public and a private network. After analyzing the graphs, the weaknesses of the network and unidentified devices in the network are identified, creating the possibility of taking appropriate measures to protect against cyber-attacks.

Keywords: network traffic, NetworkX, malicious device.

1. Introduction
Technology continues to advance at a rapid pace. Moreover, accessing the Internet has become easier, so almost everyone uses websites and web applications. Large amounts of data are stored in data centers around the world, which contain personal information, financial details and other sensitive information. Storing large amounts of data has made security threats more advanced and complex, such as viruses, malware attacks, phishing, ransomware, etc.

Hence there is an increased need to protect data against various threats as a data breach or theft can cause billions of dollars in losses to companies and can have other devastating consequences for people. The Python programming language is a very flexible tool to quickly and easily develop custom scripts to test and find potential network or system vulnerabilities.

Python is widely used by security programmers because it uses a large number of libraries to quickly develop cybersecurity applications, as well as to perform security testing, detection, and analysis of cyber threats. Also researchers and ethical hackers use Python libraries. Some of the libraries used are Scapy, Requests/Beautiful Suup, Impackets, Libmap/Nmap. With an increase in the severity of cyber attacks, innovation and advancement in the field of cyber security has become mandatory. Python is a versatile

language, with which scripts can be created to take as many measures as possible to guarantee the complete security of the system.

Scripts are also created in Python to scan ports and compromised network devices. For this article, two sets of concrete data were obtained from the scanning of a public and a private network.

After analysis, network weaknesses, unidentified devices or suspicious devices are identified, creating the possibility of taking appropriate measures to protect against cyber attacks.

2. The overall purpose of the paper

The purpose of this article is to show that the use of Python programming language scripts is quite practical and useful to find exploits, threats, devices and harmful communications in small networks of a private environment or in a public environment.

To achieve this goal, the Wireshark tool was used to analyze the public network and the private network. It resulted in the most detailed datasets for network devices and communications between them. Using Python libraries and scripts, it was identified from the dataset which of the devices communicates with the most and which of the "source" and "destination" devices for both networks, to identify suspicious or unknown devices.

Also, by means of Python scripts, the communication protocols were identified through which the devices communicate with each other, in order to find unknown devices that transmit harmful information. Using the Python script, a graph with colored nodes was created for each network to identify 'malicious' devices and their connections with other devices so that appropriate measures can be taken to secure the network.

So, through the two examples, it was best demonstrated that Python scripts are very effective and fast for network security testing.

3. Using the Python Language for Cyber Security

Python is a useful programming language for cybersecurity professionals because it performs many cybersecurity functions, such as malware analysis, port scanning, network scanning, etc. Python is often recommended as the first language new cybersecurity professionals should learn because of its widespread use and ease of learning.

Python is the top programming language in TIOBE (17.18%, Nov 2022) [1] and in PYPL (28.34%, Dec 2022)[2]. Python is the most popular language, it grew the most in the last 5 years (8.7%) [2].

Python for cybersecurity is useful because of its extensive libraries. Python has many libraries and frameworks which are very much useful and make work of a cybersecurity professional easier.

The flexibility and ease of use of Python makes it a useful tool for cyber security. Cyber professionals program the solutions they need quickly and with fairly simple code, as errors in the code are easy to find and correct.

Cybersecurity professionals can perform any task that requires them to use Python code. They use the WireShark tool to list communications between devices, and Python libraries that take these datasets as input, to analyze network traffic and identify suspicious devices. Some of these libraries are networkX, matplotlib, etc.

Below we will see how to use the networkX library to identify suspicious devices.

4. Network traffic analysis with Python scripts

Devices connected to the network, among other things, keep detailed information about their activity. They store information about the devices with which it communicates, the files that are downloaded or even communication protocols. Such information is very important to record and then to eliminate any threat that may come from harmful devices.
If on an attacked computer, we analyze its activity with Python scripts, malicious applications or files are identified. To list the communications between the devices, the WireShark tool was used, which results in a dataset file. As an example, for network traffic analysis, a public network and a private network were taken.

To analyze the dataset of the private and public network, some of the Python libraries [3] were used, such as:

- Pandas nor Pd, receives the data and displays it in the form of a Dataframe,
- MatPlotlib as plt, presents the data in graphic form,
- Networkx as nx, presents the data in the form of a network with the respective connections as nodes.

4.1. Private network and public network traffic analysis

By using a Python script, it can be identified which of the devices communicates more and with which less of the "source" devices. The same logic is followed for the "destination" devices.

The output from the script using the private network dataset shows that the IP with the most communication from the source device is 192.168.5.u, specifically 13 communications sent and that the destination device with the most communications is 239.255.255.v with 14 communications received.

While the output from the script using the public network dataset shows that the IP with the most communication from the source device is 192.168.88.x, specifically 470 communications sent and from the destination device is the device with IP 192.168.88.x with 489 communications received, so it is a known access point device.

To preserve confidential data, the last IP number is marked with letter or is deleted, since the purpose of the article is to show how with python code we can perform security tests and the data used for this purpose are illustrative.

From the analysis of both networks, no suspicious or unknown devices are identified.

Script 1. Obtaining the table showing the source:

```
import pandas as pd
import networkx as nx
import matplotlib.pyplot as plt
df = pd.read_csv('data2.csv')
print(df)
sources= df.groupby("Source").Source.count()
a=sources.sort_values()
print(a)
```

Output:

Table 1. Source of private network

IP	No.of communication
192.168.1.a	1
192.168.5.b	1
192.168.5.c	1
138.199.36.d	3
192.168.5.e	4
192.168.5.f	4
192.168.1.g	12
192.168.5.u	13

Source: Authors

Script 2. Obtaining the table showing the destination of the devices.

```
destinations=df.groupby("Destination").Source.count()
b=destinations.sort_values()
print(b)
```

Output:

Table 2. Destinacion of public network

IP	No.of communication
239.255.102.q	1
138.199.36.m	3
192.168.5.n	3
224.0.0.n	5
224.0.0.p	13
239.255.255.v	14

Source: Authors

During the analysis of the private network dataset, the communication protocols through which the devices communicate with each other are also identified, since some devices may be unknown, which means that there are events that are harmful and transmit harmful information.

Below, the Python script is used to group the data according to the communication protocols, to demonstrate how secure the information transmitted in the private network is.

It is evident that in the two networks taken as an example, the communication protocols issued by the Python script are known.

Script 3. Obtaining the table showing the type of communication protocol

```
protocols=df.groupby("Protocol").Source.count()
c=protocols.sort_values()
print(c)
```

Output:

Table 3. Protocols of private network

Protocol	No.of protocols
IGMPv2	4
MDNS	6
TCP	6
ARP	7
LLMNR	13
SSDP	13

Source: Authors

The source IP, destination IP and protocols extracted by the Python script from the public network dataset are not placed in this article as they took up a considerable space, but the analysis of the results is done in the same way as the analysis of the private network.

5. Creating colored graphite of networks to identify 'malicious' devices

After the data has been imported and we have made the relevant analysis regarding the source, destination and type of communication protocol, we can go further by presenting the data in the form of a graph [3]. Respectively, the graphite nodes will be the devices and the ribs will be the connections that each of the devices forms with the other devices. This is done in Python with the help of the networkX library, which takes the data in the form of a dataframe and turns it into a graph to see their connections in a schematic form and clearly. This action was taken for both networks considered, respectively private and public.

Script 4. Graph extraction from Dataframe data

```
network=nx.from_pandas_edgelist(df,source='Source',target='Destination',edge_attr=True)
d=network.nodes()
c=network.edges()
print(d)
print(c)
nx.draw_circular(network, with_labels=True)
plt.show()
```

Output:

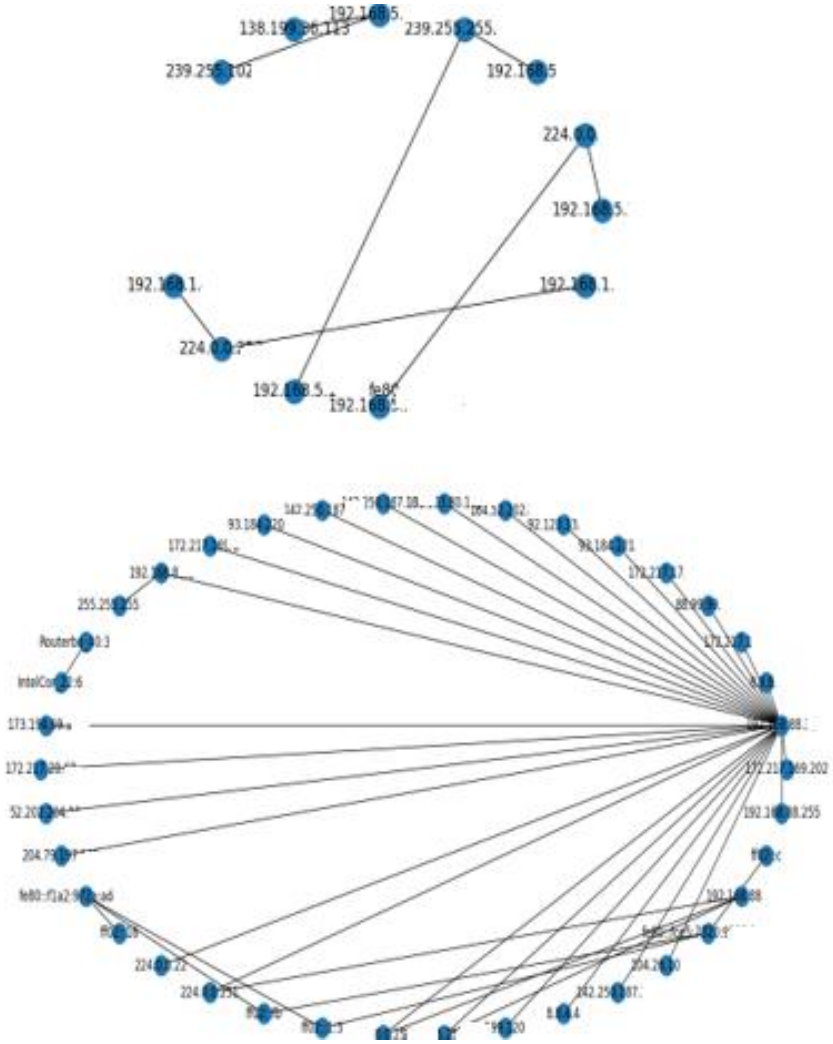

Fig. 1. Graph obtained with dataframe data, private (left), public (right))
Source: Authors

If we assume a suspicious and unknown device, which we distinguish with another color (red) and see the connections it creates with other devices, respectively with each of them presented in green [3].

By taking a deeper look at the suspected device, we can make a new data frame that includes other devices it communicates with, the type of communication, when the communication occurs, and information about that communication. In this way, we identify the harmful devices and the devices with which they communicate and then take the appropriate measures to secure the network.

Script 5. Obtaining colored graphite with suspected harmful equipment

```
suspect="192.168.88.x"
pos=nx.spring_layout(network)
nx.draw(network,pos,node_color="green",node_size=300,with_labels=True)
options={"node_size":1000,"node_color":"r"}
nx.draw_networkx_nodes(network,pos,nodelist=[suspect],**options)
plt.show()
f=df.loc[df["Source"]=='192.168.5.u']
print(f)
```

Below is the colored mesh of the private and public network.

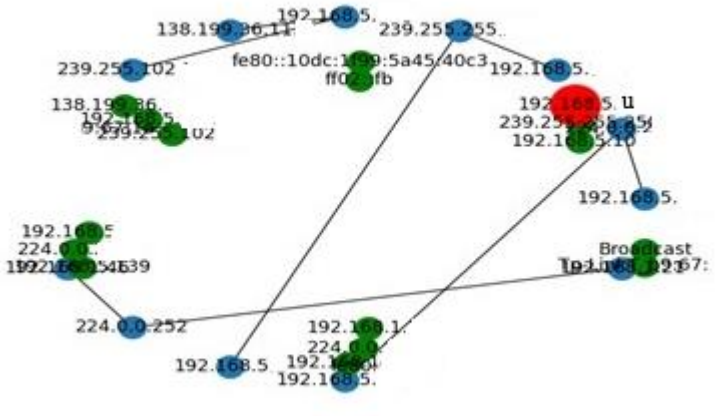

Fig. 2. Creating colored graffiti of the private network to identify a suspected malicious device
Source: Authors

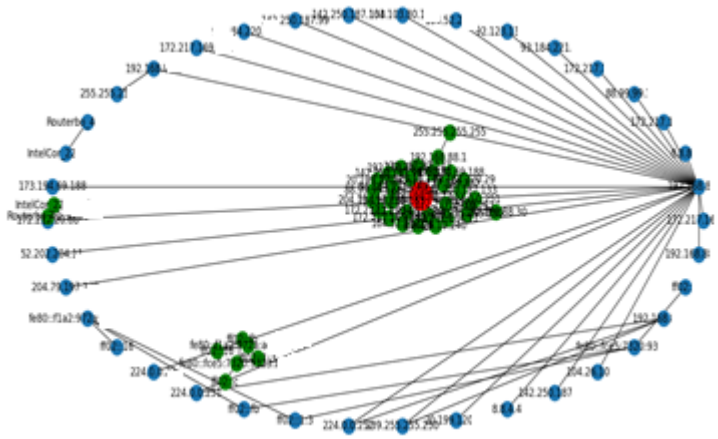

Fig. 3. Creating colored graffiti of the public network to identify a suspected malicious device
Source: Authors

Looking at our results we see which devices it communicates with and other devices it could have potentially reached, which are shown in green.

6. Conclusion

The Python language, especially in recent times, has received a rapid development and use. This language is mainly used in cyber security because of the powerful libraries it provides. Python represents an algorithm-oriented language [4] that is widely used in the field of cyber security. The above application had to do with network traffic analysis which is one of the many reasons that prove that Python enables the development of flexible and functional applications that deal with most of today's cyber security issues. With a versatile language like Python, many measures can be taken to ensure complete system and network security.

With the inevitable fact that technology and threats are constantly evolving, if professional cybersecurity skills do not envolve with them, these technologies will become ineffective and irrelevant to the organizations that use them [4]. It is therefore suggested that cyber security professionals add to their professional skills and knowledge in the use of the Python programming language to create scripts that are fast and efficient in identifying security issues.

From this article it turned out that Python scripts are quite practical and fast in identifying unknown, harmful devices or communications with harmful information.

References

[1] TIOBE Programming Community index, https://www.tiobe.com/tiobe-index/, date: 02.12.2022.

[2] PYPL PopularitY of Programming Language index, https://pypl.github.io/PYPL.html, date: 02.12.2022..

[3] Cortez, A. (2022), *Network Traffic Analysis with Python*, https://python.plainenglish.io/network-traffic-analysis-with-python-f95ed4e76c28, date, 01.12.2022.

[4] Kuk, K., Petar, M., Spalevi, M., & Goci, M. (2019), *Algorithm design in Python for cybersecurity*, Electrotechnical and Computer Science Conference, ERK, Slovenia, https://www.researchgate.net/publication/336406416.