Centre For Cybersecurity

Python Fundamental (Eddie) CFC2407

Objective

Create automation to display the operating system information.

- **Displaying the OS version**
- **Displaying the private IP address, public IP address, and the default gateway**
- **Display the hard disk size; free and used space**
- **Display the CPU usage**
- **Display the directories and their size**

Display the OS version

A module contains executable statements as well as function definitions. These statements are intended to initialize the module.

```
6   import platform
7   import requests
8   import os
9   import psutil
```

- Platform module
The platform module provides functions that access information of the underlying platform (operating system).

- Requests module
The requests module allows you to send HTTP requests. As a result, the HTTP request returns a respond object with all the response data.

- OS module
The OS module (Operating System) provides easy functions that allow us to interact and get Operating System information and even control processes.

- Psutil module
Psutil (Python system and process utilities) is a cross-platform library for retrieving information on running processes and system utilization (CPU, memory, disks, network, sensors) in Python.

We are using the platform module present in the Python standard library for gathering this basic system information.

```
import platform

# Displaying The Basic System Infomation
print("\n\t\t\t Basic System Information\n")

# Printing the Architecture of the OS
print("[+] Architecture :", platform.architecture())

# Displaying the machine
print("[+] Machine :", platform.machine())

# Printing the Operating System release information
print("[+] Operating System Release :", platform.release())

# Prints the currently using system name
print("[+] System Name :",platform.system())

# This line will print the version of your Operating System
print("[+] Operating System Version :", platform.version())
```

In the above code, we first imported the platform module. We use the platform module's functions to get the required information.

```
┌──(eddie㉿kali)-[~/forloop]
└─$ python3 pyeddie.py

                        Basic System Information

[+] Architecture : ('64bit', 'ELF')
[+] Machine : x86_64
[+] Operating System Release : 5.18.0-kali5-amd64
[+] System Name : Linux
[+] Operating System Version : #1 SMP PREEMPT_DYNAMIC Debian 5.18.5-1kali6 (2022-07-07)
```

As we can see in the output above, the program displays many important details about the operating system like system architecture, platform and etc.

Display the private IP address, public IP address, and the default gateway

We can filter the internal IP address and the default gateway using the OS module (os.system) by using the arguments as how we would use it to filter the internal IP address and the default gateway on the Linux terminal. Do note that the arguments need to be stored in a variable in order for it to work.

Next, we would be using the request module (request.get) to retrieve the public IP address from a third party website.

```
29  import os
30  import requests
31
32  # Displaying The Network Information
33  print("\n\t\t\t Network Information\n")
34
35  # Storing the arguments that filter out the internal IP address in a variable "ip"
36  ip="(ifconfig | grep broadcast | awk '{print $2}')"
37
38  # Using the OS module to print out the internal IP address with the variable "ip"
39  print("[+] Internal IP address :",os.system(ip))
40
41  # Using the request module to retrieve public IP 'https://api.ipify.org' and storing in variable
42  pubip = requests.get('https://api.ipify.org').text
43
44  # Print out the  Public IP Address with the variable "pubip"
45  print("[+] Public IP address :", pubip)
46
47  # Storing the arguments that filter out the default gateway in a variable "gateway"
48  gateway="(route | grep UG| awk '{print$ 2}')"
49
50  # Using the OS module to print out the default gateway with the variable "gateway"
51  print("[+] Default Gateway :" ,os.system(gateway))
```

Below is the output after running the code.

```
┌──(eddie㉿kali)-[~/forloop]
└─$ python3 pyeddie.py

                        Network Information

192.168.149.130
[+] Internal IP address : 0
[+] Public IP address : [redacted]
192.168.149.2
[+] Default Gateway : 0
```

Display the hard disk size; free and used space

Just like the private IP address we can use the OS module (os.system) to get the system disk details by using the arguments.

```
51  import os
52
53  # Displaying Disk Details
54  print("\n\t\t\t Disk Details\n")
55
56  # Storing the arguments that retrieve the system disk details
57  disk_details='(df -h)'
58
59  # Using the OS module to print out the disk details with the variable "disk_details"
60  print(os.system(disk_details))
```

As we can see below, the disk details are listed down for the various file systems after running the codes.

```
┌──(eddie㉿kali)-[~/forloop]
└─$ python3 pyeddie.py

                        Disk Details

Filesystem      Size  Used Avail Use% Mounted on
udev            950M     0  950M   0% /dev
tmpfs           198M  1.2M  197M   1% /run
/dev/sda1        79G   14G   61G  19% /
tmpfs           987M     0  987M   0% /dev/shm
tmpfs           5.0M     0  5.0M   0% /run/lock
tmpfs           198M  4.8M  193M   3% /run/user/1001
0
```

Display the current CPU usage

We area using the psutil module (psutil.cpu_freq()) to print out the current CPU usage.

```
62  import psutil
63
64  # Displaying The CPU Usage
65  print("\n\t\t\t CPU Usage\n")
66
67  # Using the psutil module to print out the current CPU usage
68  cpu_frequency = psutil.cpu_freq()
69  print("[+]Current Frequency :", cpu_frequency,"GHz")
70
```

As we can see below, the code prints out the current CPU usage.

```
┌──(eddie㉿kali)-[~/forloop]
└─$ python3 pyeddie.py

                        CPU Usage

[+]Current Frequency : scpufreq(current=2903.999, min=0.0, max=0.0) GHz
```

Displaying all the directories in the current directory

First, we will be storing the current directory in a variable (dir_items). Then we will be running the forloop for each item in the directory and store the size of the items using the ('os.path.getsize()') in a variable ('filesize').

Next, use the IF statement to check if the item in the list is a directory. And if it is true, it will only print out the item which is a directory with its size. Do note that the sizes are in bytes format.

```
71  import os
72
73  # Displaying The Current Directory
74  print("\n\t\t\t Current Directories\n")
75
76  # Using forloop to print out the only the diretories with its size
77
78  # Listing out the contents in the directory and saving it in a variable
79  dir_items=os.listdir()
80
81  # Running forloop
82  for eachitem in dir_items:
83
84  # Saving the items size in the list in a variable
85      filesize=os.path.getsize(eachitem)
86
87  # Checking for the the list for directories only
88      if os.path.isdir(eachitem)==True:
89
90  # Printing the directory name and size
91          print(eachitem," ",os.path.getsize(eachitem),"bytes")
```

Here is the output after running the codes above. We also can see that the item highlighted in blue is a directory to prove that the output is only the directories.