

CYBR 493A

Introduction

Chapter 3: Ping in Python
text: chapter 3 of Python for Cybersecurity

TOPICS

- Importing modules in Python
- Creating scripts to ping devices in a network
- Pause: Installing git on Ubuntu
- Python Conditionals
- Extending our script to work on multiple Operating Systems
- Python Loops
- Using loops to ping multiple hosts
- Python Functions
- Ping function

Introduction

- We can import pre-defined modules, libraries, and packages in python.
- Programmer and developers do not write everything they need from scratch
- They import codes that were developed by others.
- Then, we can simply use the capabilities of these modules.

Importing modules into Python

- Before we can import any modules, we need to make sure they are installed/ available in our miniconda environment.
- Navigate to your miniconda prompt and activate the environment you are wish to use.

Make sure you run as administrator

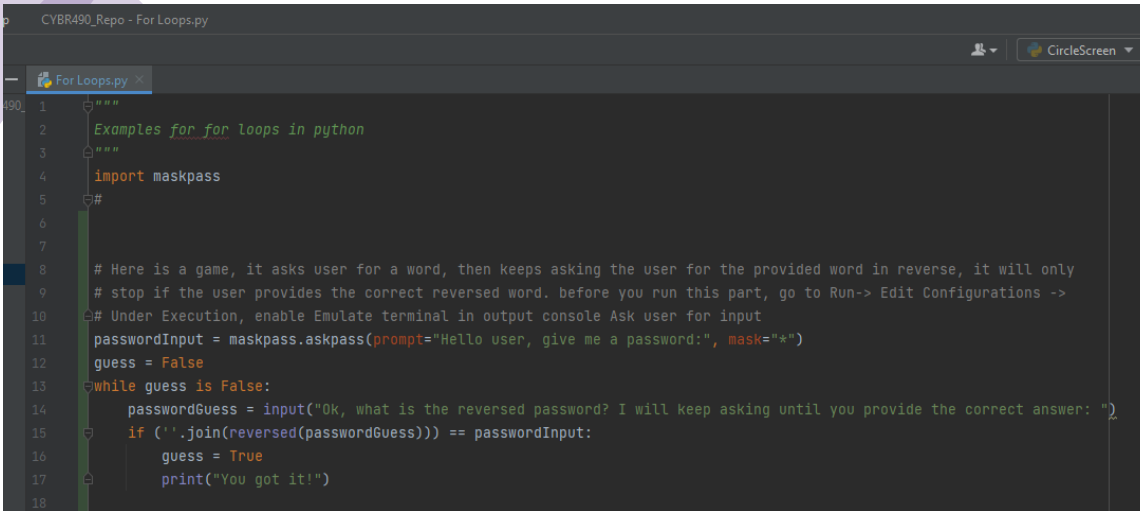
- Then install the module (e.g., maskpass) type this command:

```
>> pip install [module name]
```

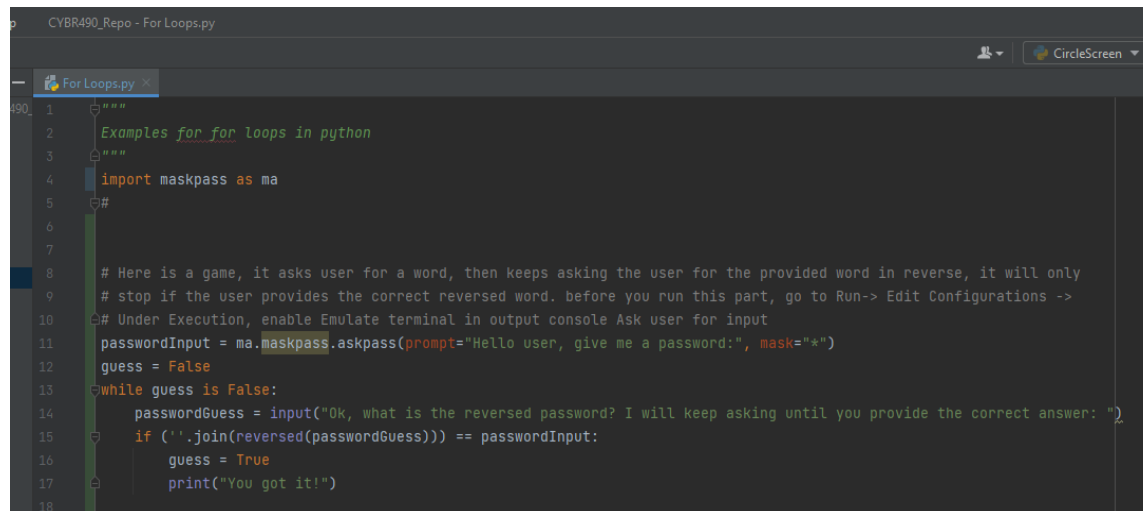
Importing modules into Python

- After installing the module, you may import it from your pycharm using the command:
 - Import [package name].
 - If you see redline under the module name, this means one of two things:
 - You have not installed the module
 - You have installed it in a different miniconda environment.

Importing modules into Python



```
1  """
2  Examples for for loops in python
3  """
4  import maskpass
5  #
6
7
8  # Here is a game, it asks user for a word, then keeps asking the user for the provided word in reverse, it will only
9  # stop if the user provides the correct reversed word. before you run this part, go to Run-> Edit Configurations ->
10 # Under Execution, enable Emulate terminal in output console Ask user for input
11 passwordInput = maskpass.askpass(prompt="Hello user, give me a password:", mask="*")
12 guess = False
13 while guess is False:
14     passwordGuess = input("Ok, what is the reversed password? I will keep asking until you provide the correct answer: ")
15     if ''.join(reversed(passwordGuess)) == passwordInput:
16         guess = True
17     print("You got it!")
18
```



```
1  """
2  Examples for for loops in python
3  """
4  import maskpass as ma
5  #
6
7
8  # Here is a game, it asks user for a word, then keeps asking the user for the provided word in reverse, it will only
9  # stop if the user provides the correct reversed word. before you run this part, go to Run-> Edit Configurations ->
10 # Under Execution, enable Emulate terminal in output console Ask user for input
11 passwordInput = ma.maskpass.askpass(prompt="Hello user, give me a password:", mask="*")
12 guess = False
13 while guess is False:
14     passwordGuess = input("Ok, what is the reversed password? I will keep asking until you provide the correct answer: ")
15     if ''.join(reversed(passwordGuess)) == passwordInput:
16         guess = True
17     print("You got it!")
18
```

- An optional parameter is to assign a name to the module.
- See the two screenshot.
- On the top screenshot, we imported maskpass and were able to use it by its name directly in line 11.
- On the bottom screenshot, we gave maskpass a name (ma). And we can use the maskpass built-in function by using the name ma, instead of the name of the whole module.

Pinging devices in a network

- As with all things related to computers, there are several ways to perform the same task. One example is the process of pinging an address.
- Ping is a method of sending a message to a remote computer, and then having them respond back.
- This is useful for security and network engineers to find out which resources are online

Note before we start

- Remember that we may run python codes from several places:
 1. PyCharm: write your entire code and run all of it at once
 2. Jupyter Notebook: write each functionality of your code and then run it.
 3. Python interactive mode: similar to Jupyter, but without the ability to save.
 - Activate miniconda environment, then type python and hit enter. You are now in the Python interactive mode.

Pinging in Python

- Lets us first try the command by pinging the default loopback address 127.0.0.1.
- Syntax:

```
os.system ("ping [options] [ip  
address] )
```

- You will need to import platform and os

Pinging in Python

```
Administrator: Anaconda Prompt (Miniconda3) - python
(base) C:\WINDOWS\system32>activate pythonForCyber

(pythonForCyber) C:\WINDOWS\system32>python
Python 3.9.12 (main, Apr  4 2022, 05:22:27) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import platform
>>> import os
>>> os.system("ping -c 1 -w 2 127.0.0.1")

Pinging 127.0.0.1 with 32 bytes of data:
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128

Ping statistics for 127.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
0
>>> _
```

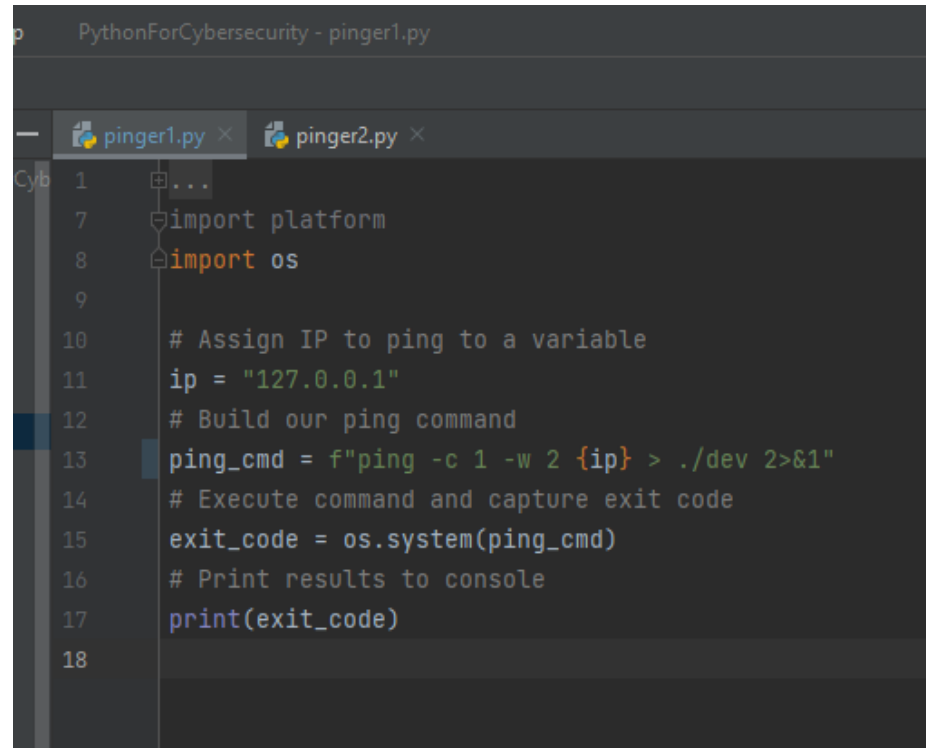
- The number 0, the exit code, at the end shows the command is successfully executed, otherwise it will be something else.
- The first option (-c) allows us to specify the number of packets, or attempts, to send to the target.
- The second option (-w) allows us to specify how long we should wait for a response.

Pinging in Python

A screenshot of an Anaconda Prompt window titled "Administrator: Anaconda Prompt (Miniconda3) - python". The window has a black background with yellow text. On the left side, there are several small, light blue icons representing different file types or folders. The main area shows a series of prompt characters ">>>" followed by a command: `os.system("ping -c 1 -w 2 127.0.0.1 > c:\dev 2>&1")`. Below this command, there is a single character "0" and another prompt character ">>>". At the bottom of the window, there is a red text string: `> .\dev\pull_2>84"`.

- We can modify ping to a binary response (failure or success)
- Add > location followed by c:\dev and 2>&1

First Ping

A screenshot of a code editor window titled 'PythonForCybersecurity - pinger1.py'. The editor shows a Python script with line numbers 1 through 18. The script imports 'platform' and 'os', assigns the IP '127.0.0.1' to a variable 'ip', builds a ping command string, executes it using 'os.system()', and prints the exit code. The script is as follows:

```
1 ...  
7 import platform  
8 import os  
9  
10 # Assign IP to ping to a variable  
11 ip = "127.0.0.1"  
12 # Build our ping command  
13 ping_cmd = f"ping -c 1 -w 2 {ip} > ./dev 2>&1"  
14 # Execute command and capture exit code  
15 exit_code = os.system(ping_cmd)  
16 # Print results to console  
17 print(exit_code)  
18
```

- Notice what happens after you run this code.

Let us retake a look at Linux

- Download and install Oracle Virtual Box
- Download a copy of Kali Linux
- Add Ubuntu (or Kali) Virtual Machine:
 - <https://ubuntu.com/tutorials/how-to-run-ubuntu-desktop-on-a-virtual-machine-using-virtualbox#1-overview>
 - <https://www.youtube.com/watch?v=x5MhydiJWmC>

Designate a folder for repositories

- Let us create a new folder, in the terminal type:
- `>> mkdir [git_repos]`
- Then, cd to your repo.

Clone repo in Linux

- In the terminal, type:

```
>> git clone [url from github.com]
```

- View all files in repo:

```
>> ls
```

- OR

```
>> ls -l
```

Run Python Files

- In the miniconda environment, run:

```
>> python [python file name.py]
```