



Ping , Telnet and SSH with python

Content

1. ping with **os**
2. telnet with **telnetlib**
3. SSH with **paramiko**

Python

1.

Ping

Ping คืออะไร

- ระบบทดสอบคอมพิวเตอร์ปลายทางว่าเชื่อมต่อกับ "ระบบ IP" หรือไม่

```
C:\>ping -t 192.168.1.33
```

```
Pinging 192.168.1.33 with 32 bytes of data:
```

```
Reply from 192.168.1.33: bytes=32 time=-33ms TTL=128  
Reply from 192.168.1.33: bytes=32 time=-33ms TTL=128  
Reply from 192.168.1.33: bytes=32 time=33ms TTL=128  
Reply from 192.168.1.33: bytes=32 time=-33ms TTL=128  
Reply from 192.168.1.33: bytes=32 time=-33ms TTL=128  
Reply from 192.168.1.33: bytes=32 time=33ms TTL=128  
Reply from 192.168.1.33: bytes=32 time=33ms TTL=128
```

***Ping** With python*

```
import os
```

- os เป็น library พื้นฐานของ python อยู่แล้ว
- เมื่อทำการเขียนไฟล์ python สามารถประกาศเรียกใช้ os ได้ทันที

Ping syntax

```
os.system("ping " + hostname)
```

Syntax การ ping ด้วย os

Example

```
import os
hostname = "google.com"
response = os.system("ping " + hostname)
print response
if response == 0:
    print hostname, 'up'
else:
    print hostname, 'down'
```

Output

0

google.com up

ความหมาย Response

0 คือ ping เจอ

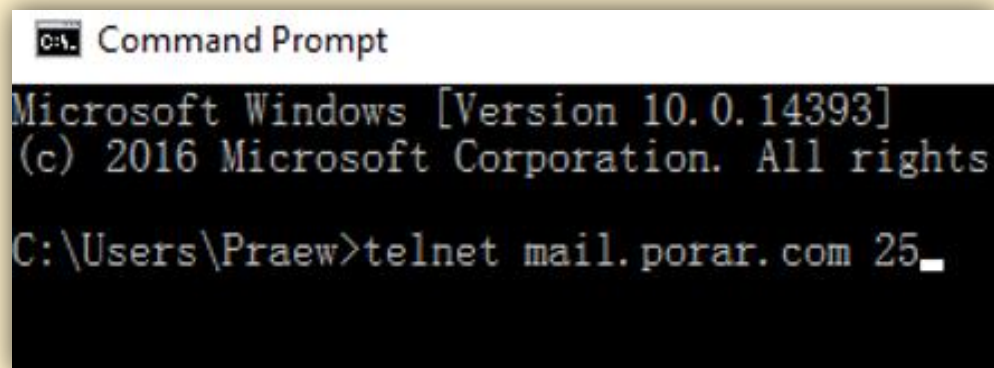
นอกเหนือจากนี้คือ ping ไม่เจอ

2.

telnet

Telnet คืออะไร

- เป็นหนึ่งในวิธีการควบคุมคอมพิวเตอร์เครื่องอื่นจากระยะไกล
- เข้าถึงบริการ ข้อมูล ของคอมพิวเตอร์เครื่องนั้นได้
- เป็นวิธีที่ดีในการตรวจสอบ port ของเครื่องคอมพิวเตอร์เป้าหมายว่าสามารถเข้าถึงและใช้งาน
ได้หรือไม่



```
C:\> Command Prompt
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\Praew>telnet mail.porar.com 25_
```

Telnet with python

```
import telnetlib
```

- telnetlib เป็น library พื้นฐานของ python อยู่แล้ว
- เช่นเดียวกับ os เราสามารถประกาศเรียกใช้ telnetlib ได้ทันทีเมื่อเขียนไฟล์ python

*How to use **Telnet***

```
tn = telnetlib.Telnet(HOST,PORT)
```

HOST = IP ของเครื่องคอมพิวเตอร์ที่เราต้องการ Telnet

PORT = หมายเลข PORT ของคอมพิวเตอร์ที่เราต้องการเข้าถึง

Example

Input 1

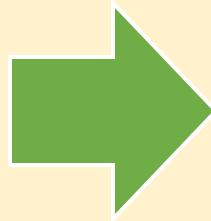
HOST : "8.8.8.8"

PORT : 80

Input 2

HOST : "203.151.223.57"

PORT : 5000



```
import telnetlib
```

```
try:
```

```
    tn = telnetlib.Telnet(HOST,PORT,timeout=2)
```

```
    print "telnet success"
```

```
    tn.close()
```

```
except:
```

```
    print "telnet fail"
```

OUTPUT

1.) "telnet success"

2.) "telnet fail"

exercise

ให้ทำการ ping IP และ Domain ต่อไปนี้

- 203.161.227.51
- google.com
- inet.com
- 203.151.223.57

หาก ping IP นั้นพบ ให้ telnet ไปยัง port ดังนี้

- 9999 , 443

ทำการแสดงผลลัพธ์ตามรูปแบบ OUTPUT ด้านข้าง

OUTPUT

>>> IP 203.154.58.132 <<<

ping : not found

telnet: not doing

>>> IP 203.154.58.132 <<<

ping : found

telnet:

9999 -> success

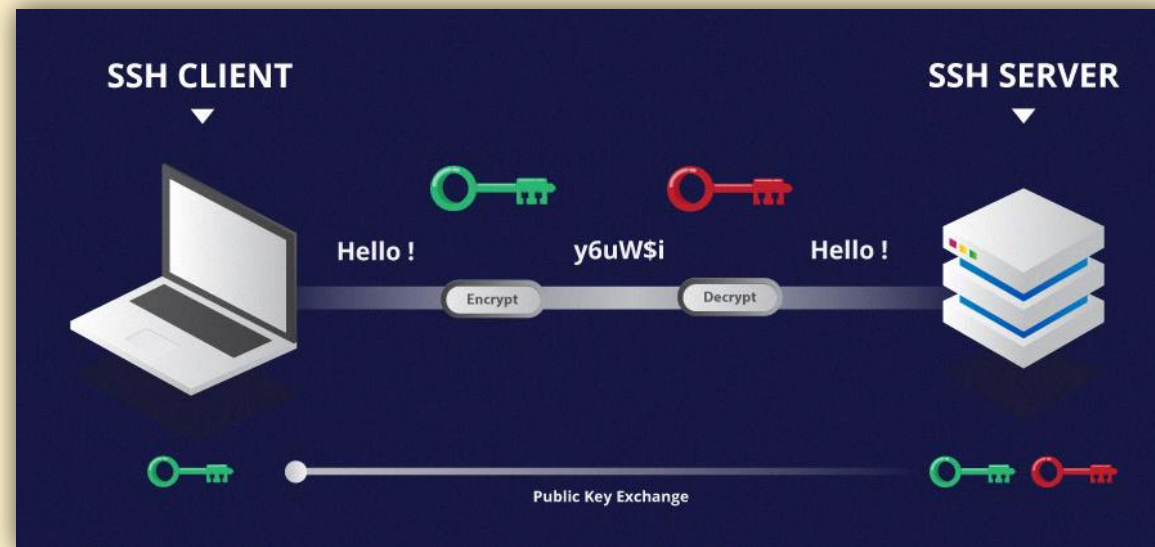
443 -> fail

3.

SSH

What the *SSH* ?

- Secure Shell
- หนึ่งในวิธีเข้าถึงคอมพิวเตอร์เครื่องอื่น และเข้าควบคุมจากระยะไกล
- วิธีการนี้มีการใช้วิธีเข้ารหัสข้อมูลที่ส่งผ่านระหว่างเครื่อง
- ถูกออกแบบมาเพื่อทดแทน telnet, Rlogin, RSH ด้วยเหตุผลทางด้านความปลอดภัย

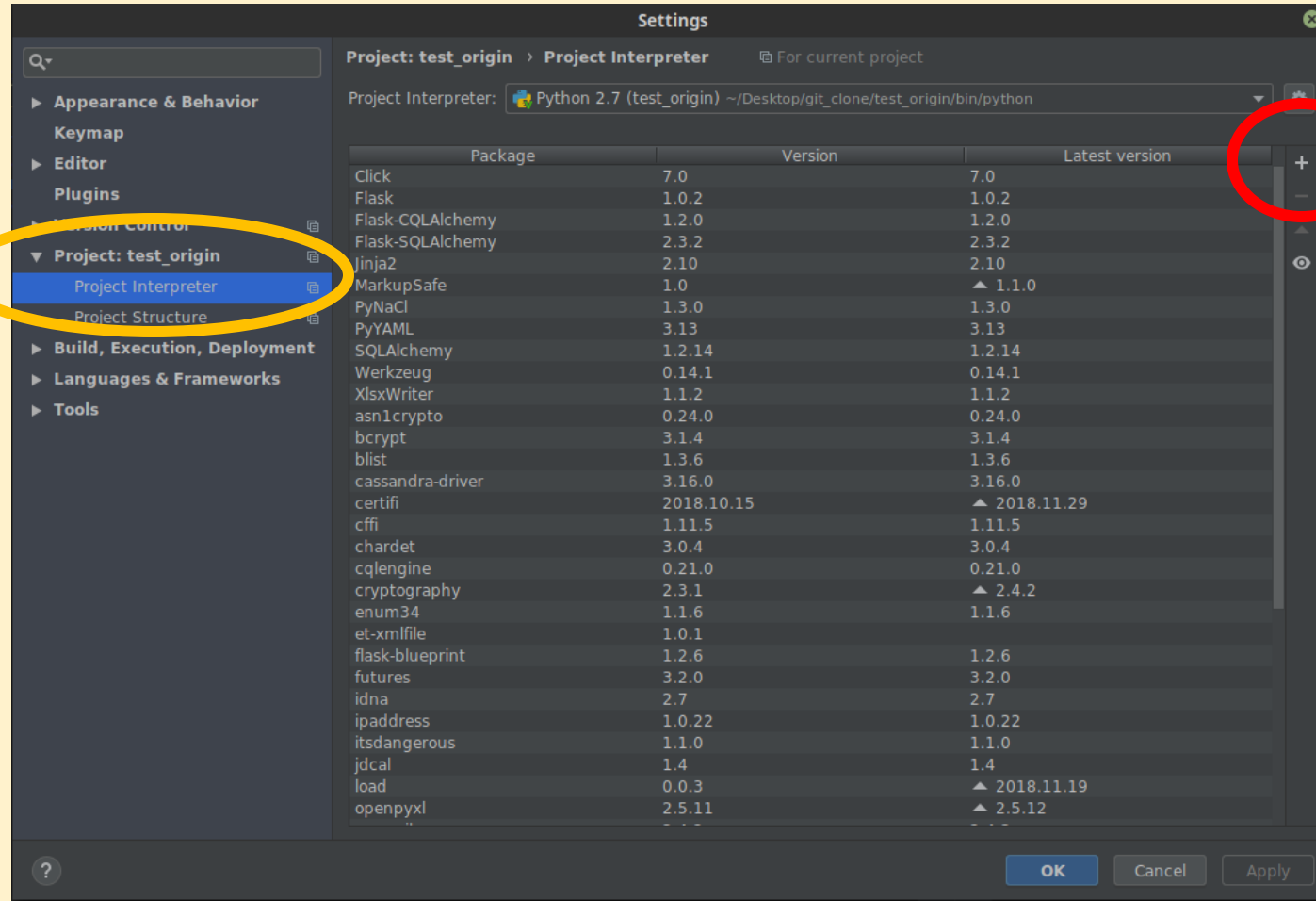


```
import paramiko
```

- ต้องทำการ install เพิ่มเติมบนเครื่องคอมพิวเตอร์ก่อน
- หากติดตั้งเสร็จแล้ว เมื่อเขียนไฟล์จะสามารถประกาศเรียกใช้งาน paramiko ได้

INSTALL paramiko (สำหรับคนที่ใช้ pycharm)

2.) หัวข้อ
Interpreter



3.) กด
เครื่องหมาย
+

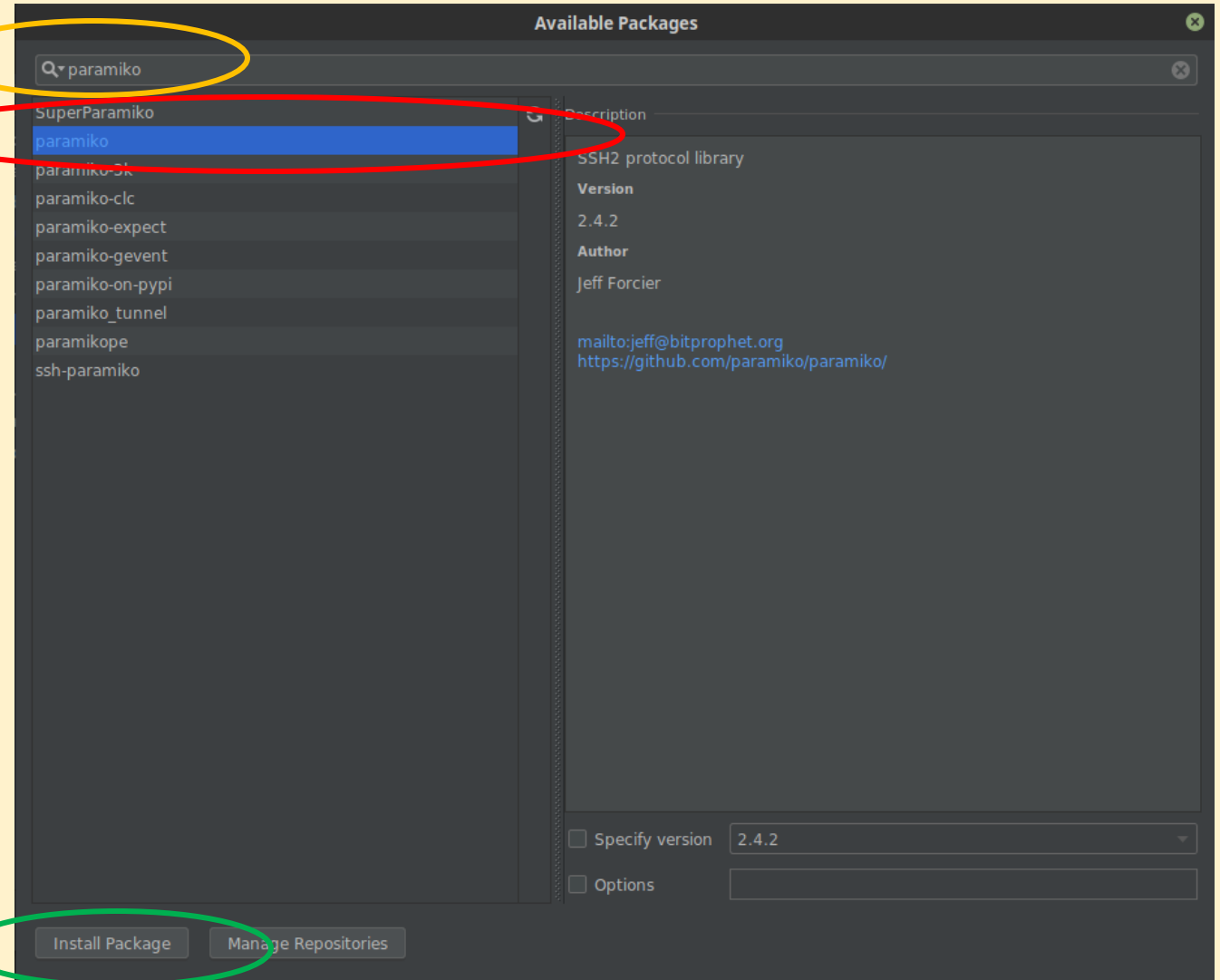
1.) Ctrl + S

INSTALL paramiko (สำหรับคนที่ใช้ pycharm) (ต่อ)

1. พิมพ์ค้นหา paramiko

2. เลือกผลลัพธ์การค้นหาชื่อ
"paramiko"

3. กดปุ่ม install



INSTALL paramiko **(command)**

```
pip install paramiko
```

Command ในการ install paramiko

INSTALL paramiko **(command)** **(ต่อ)**

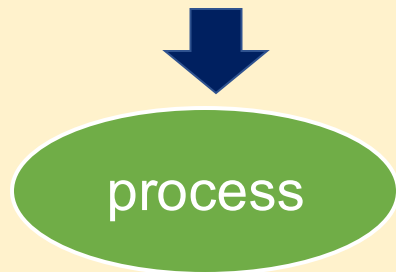
```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.

C:\Users\Administrator>cd ..
C:\Users>cd ..
C:\>cd python27
C:\Python27>cd Scripts
C:\Python27\Scripts>_
```

1. เข้าใช้งาน Windows Command interpreter (CMD)
2. เข้าไปยัง folder : Python27/Script
3. พิมพ์คำสั่ง pip install paramiko

Syntax Paramiko

```
client = paramiko.SSHClient()  
client.set_missing_host_key_policy(paramiko.AutoAddPolicy())  
client.connect(hostname=HOST, username=USER, password=PASS, key_filename=KEY )
```



client.close()

HOST = IP เครื่องเป้าหมาย

USER = ชื่อบัญชีผู้ใช้สำหรับเข้าใช้งาน

PASS = รหัสผู้เข้าใช้งาน

KEY = ไฟล์ key เพื่อขอเข้าถึง HOST นี้

Example host

HOST

35.243.183.158

username

testssh

password

mis@Pass01

Insert Command

```
client.connect(hostname, port=port, username=username, password=password)
stdin, stdout, stderr = client.exec_command(command)
print stdout.read()
client.close()
```

Command = คำสั่งที่ใช้บนคอมพิวเตอร์เป้าหมาย

เช่น "ls" , "df -h" , "whoami"

Insert Command (ต่อ)

```
client = paramiko.SSHClient()
client.set_missing_host_key_policy(paramiko.AutoAddPolicy())
client.connect(hostname="35.243.183.158", username="testssh",
password="mis@Pass01")
stdin, stdout, stderr = client.exec_command("whoami")
print stdout.read()
client.close()
```

OUTPUT

testssh

Multiple Command

INPUT

"""

whoami\n

ls\n

cd folder_01\n

pwd\n

ls\n

"""

```
stdin, stdout, stderr = client.exec_command(INPUT)
print stdout.read()
client.close()
```

OUTPUT

testssh

folder_01

folder_02

/home/testssh/folder_01

folder_test

exercise

- ทำการ SSH เพื่อดึงค่าข้อมูลจากเครื่องเป้าหมาย "35.243.183.158"
- ข้อมูลที่ต้องการทราบคือ
 1. พื้นที่ของ host ที่ถูกใช้งานแล้ว
 2. พื้นที่ของ host ที่เหลือว่าง
 3. จำนวนพื้นที่ทั้งหมดของ host
- แสดงผลลัพธ์ เป็นข้อความ ดังตัวอย่าง

HINT : คำสั่งดูพื้นที่ของ HOST คือ "df"

OUTPUT

พื้นที่ถูกใช้งานไปแล้ว : XXXX (หน่วย GB)

เหลือพื้นที่ว่างอีก : XXXX (หน่วย GB)

พื้นที่ทั้งหมดของ HOST : XXXX (หน่วย GB)

EPIC exercise

จากโจทย์หน้าที่แล้วให้เพิ่มเติมการกระทำดังต่อไปนี้

- ให้ผลลัพธ์ที่ได้ แจ้งเตือนบนโปรแกรม LINE ทุก ๆ 30 วินาที
- หากพื้นที่ของ HOST ถูกใช้เกิน 50% ให้แจ้งเตือนว่า "DETECT" ขึ้นในทันที และอีกทุก ๆ 5 วินาทีหากว่า % ยังไม่ลดต่ำกว่า 50%
- เมื่อเกิดการ "DETECT" แล้ว หาก % ลดลงกลับมาต่ำกว่า 50 % ให้แสดงข้อความว่า "Back to normal"



Thank you