



KỸ THUẬT LẬP TRÌNH

ThS. Bùi Việt Thắng

E: thangbv82@gmail.com

T: 0983085387



Chương 2. Kiểm thử xâm nhập (red team)

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. Hành động



1. TỔNG QUAN VỀ BÀI TOÁN TẤN CÔNG

- **Cyber Kill Chain**
- **MITRE ATT&CK**



Cyber Kill Chain

- Cyber Kill Chain** là một chuỗi các bước mô tả 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



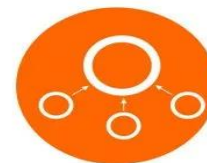
Reconnaissance



Weaponization



Delivery



Exploitation



Installation



Command & control



Actions on objectives



Cyber Kill Chain

- **Cyber Kill Chain** để giải quyết các cuộc tấn công mạng bằng cách xác định mô hình và hành vi của tội phạm mạng khi chúng thực hiện một cuộc tấn công
⇒ các quản trị viên hiểu thêm về ransomware, vi phạm bảo mật, tấn công APT, cũng như cách ngăn chặn chúng.



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 làm mục tiêu để thu thập thông tin.



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.



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 phiên, tấn công man-in-the-middle...



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.



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.



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.



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.



1. TỔNG QUAN VỀ BÀI TOÁN TẤN CÔNG

- **Cyber Kill Chain**
- **MITRE ATT&CK**



MITRE ATT&CK

- MITRE ATT & CK (MITRE Adversarial **Tactics**, **Techniques**, and Common Knowledge) là một cơ sở kiến thức và mô hình quản lý hành vi về kẻ đe dọa trên mạng, phản ánh các giai đoạn khác nhau của vòng đời tấn công của kẻ thù và các nền tảng mà chúng nhắm mục tiêu.
 - ❑ **Tactics** (Chiến thuật) biểu thị các mục tiêu ngắn hạn và chiến thuật của đối thủ trong một cuộc tấn công.
 - ❑ **Techniques** (Các kỹ thuật) mô tả cách mà đối thủ nhắm mục tiêu.
- ⇒ MITRE ATT&CK tạo ra một tài liệu toàn diện về các chiến thuật, kỹ thuật cũng như quy trình mà những kẻ tấn công mạng thường sử dụng
- ⇒ MITRE ATT&CK trở thành một cơ sở tri thức giúp tiêu chuẩn hóa an ninh phòng thủ.



ATT&CK Matrix for Enterprise

Reconnaissance TA0043	Resource Development 7 techniques	Initial Access 9 techniques	Execution 13 techniques	Persistence 19 techniques	Privilege Escalation 13 techniques	Defense Evasion 42 techniques	Credential Access 17 techniques	Discovery 30 techniques	Lateral Movement 9 techniques	Collection 17 techniques	Command and Control 16 techniques	Exfiltration 9 techniques	Impact 13 techniques
Active Scanning (5)	Acquire Infrastructure (7)	Drive-by Compromise	Command and Scripting Interpreter (6)	Account Manipulation (5)	Abuse Elevation Control Mechanism (4)	Abuse Elevation Control Mechanism (4)	Adversary-in-the-Middle (2)	Account Discovery (4)	Exploitation of Remote Services	Adversary-in-the-Middle (2)	Application Layer Protocol (4)	Automated Exfiltration (1)	Account Access Removal
Gather Victim Host Information (4)	Compromise Accounts (2)	Exploit Public-Facing Application	Container Administration Command	BITS Jobs	Access Token Manipulation (5)	Access Token Manipulation (5)	Brute Force (4)	Application Window Discovery	Internal Spearphishing	Archive Collected Data (2)	Communication Through Removable Media	Data Transfer Size Limits	Data Destruction
Gather Victim Identity Information (2)	Compromise Infrastructure (7)	External Remote Services	Deploy Container	Boot or Logon Autostart Execution (14)	BITS Jobs	BITS Jobs	Credentials from Password Stores (8)	Browser Bookmark Discovery	Lateral Tool Transfer	Audio Capture	Encrypted Channel (2)	Data Encoded (2)	Data Encrypted for Impact
Gather Victim Network Information (4)	Develop Capabilities (4)	Hardware Additions	Exploitation for Client Execution	Boot or Logon Initialization Scripts (8)	Boot or Logon Autostart Execution (14)	Build Image on Host	Exploitation for Credential Access	Cloud Infrastructure Discovery	Remote Service Session Hijacking (2)	Automated Collection	Data Obfuscation (2)	Exfiltration Over Alternative Protocol (3)	Data Manipulation (2)
Gather Victim Org Information (4)	Establish Accounts (2)	Phishing (2)	Inter-Process Communication (2)	Browser Extensions	Boot or Logon Initialization Scripts (8)	Debugger Evasion	Forced Authentication	Cloud Service Dashboard	Remote Services (8)	Browser Session Hijacking	Dynamic Resolution (2)	Exfiltration Over C2 Channel	Defacement (2)
Phishing for Information (3)	Obtain Capabilities (6)	Replication Through Removable Media	Native API	Compromise Client Software Binary	Create or Modify System Process (4)	Deobfuscate/Decode Files or Information	Input Capture (4)	Cloud Service Discovery	Replication Through Removable Media	Clipboard Data	Encrypted Channel (2)	Exfiltration Over Other Network Medium (1)	Disk Wipe (2)
Search Closed Sources (2)	Stage Capabilities (6)	Supply Chain Compromise (2)	Scheduled Task/Job (5)	Create Account (2)	Domain Policy Modification (2)	Deploy Container	Modify Authentication Process (7)	Cloud Storage Object Discovery	Software Deployment Tools	Data from Cloud Storage	Fallback Channels	Firmware Corruption	Endpoint Denial of Service (4)
Search Open Technical Databases (8)		Trusted Relationship	Serverless Execution	Create or Modify System Process (4)	Domain Policy Modification (2)	Direct Volume Access	Multi-Factor Authentication Process (7)	Container and Resource Discovery	Taint Shared Content	Data from Configuration Repository (2)	Ingress Tool Transfer	Exfiltration Over Physical Medium (1)	Inhibit System Recovery
Search Open Websites/Domains (2)		Valid Accounts (4)	Shared Modules	Event Triggered Execution (14)	Escape to Host	Execution Guardrails (1)	Multi-Factor Authentication Interception	Debugger Evasion	Use Alternate Authentication Material (4)	Data from Information Repositories (2)	Multi-Stage Channels	Exfiltration Over Web Service (2)	Network Denial of Service (2)
Search Victim-Owned Websites			Software Deployment Tools	External Remote Services	Event Triggered Execution (14)	Exploitation for Defense Evasion	Multi-Factor Authentication Request Generation	File and Directory Permissions Modification (2)		Data from Local System	Non-Application Layer Protocol	Scheduled Transfer	Resource Hijacking
			System Services (2)	Hijack Execution Flow (12)	Exploitation for Privilege Escalation	File and Directory Permissions Modification (2)	Multi-Factor Authentication Request Generation	Group Policy Discovery		Data from Network Shared Drive	Non-Standard Port	Transfer Data to Cloud Account	Service Stop
			User Execution (2)	Implant Internal Image	Hijack Execution Flow (12)	Hide Artifacts (12)	Network Sniffing	Network Service Discovery		Data from Removable Media	Protocol Tunneling		System Shutdown/Reboot
			Windows Management Instrumentation	Modify Authentication Process (2)	Process Injection (12)	Hijack Execution Flow (12)	OS Credential Dumping (2)	Network Share Discovery		Data Staged (2)	Proxy (4)		
				Office Application Startup (4)	Scheduled Task/Job (5)	Impair Defenses (9)	Steal or Forge Authentication Token	Network Sniffing		Email Collection (2)	Remote Access Software		
				Pre-OS Boot (8)	Valid Accounts (4)	Indicator Removal (3)	Steal or Forge Kerberos Tickets (4)	Password Policy Discovery		Input Capture (4)	Traffic Signaling (2)		
				Scheduled Task/Job (5)		Masquerading (7)	Steal Web Session Cookie	Peripheral Device Discovery		Screen Capture	Web Service (3)		
				Server Software Component (5)		Modify Authentication Process (7)	Unsecured Credentials (7)	Permission Groups Discovery (2)		Video Capture			
				Traffic Signaling (2)		Modify Cloud Compute Infrastructure (4)		Process Discovery					
				Valid Accounts (4)		Modify Registry		Query Registry					
						Network Boundary Bridging (1)		Remote System Discovery					
						Obfuscated Files or Information (2)		Software Discovery (1)					
						Plist File Modification		System Information Discovery					
						Pre-OS Boot (8)		System Location Discovery (1)					
						Process Injection (12)		System Network Configuration Discovery (1)					
						Reflective Code Loading		System Network Connections Discovery					
						Rogue Domain Controller		System Owner/User Discovery					
						Rootkit		System Service Discovery					
						Subvert Trust Controls (6)		System Time Discovery					
						System Binary Proxy Execution (12)		Virtualization/Sandbox Evasion (2)					
						System Script Proxy Execution (1)							
						Template Injection							
						Traffic Signaling (2)							
						Trusted Developer Utilities Proxy Execution (1)							
						Unused/Unsupported Cloud Regions							
						Use Alternate Authentication Material (4)							
						Valid Accounts (4)							
						Virtualization/Sandbox Evasion (2)							
						Weaken Encryption (2)							
						XSL Script Processing							



Cyber Kill Chain vs MITRE ATT&CK

- Cả hai đều là các framework để giải quyết các cuộc tấn công mạng nhằm vào một tổ chức.

Cyber Kill Chain	MITRE ATT&CK
Cyber Kill Chain (Chuỗi tiêu diệt mạng) giải quyết quá trình tấn công mạng từ cấp độ cao với bảy giai đoạn	MITRE ATT&CK có phạm vi kiến thức sâu hơn bao gồm các chi tiết về các cuộc tấn công mạng, chẳng hạn như thủ tục và kỹ thuật tấn công cũng như liên kết đến các khuyến nghị.
Giới thiệu cơ bản về hành vi tấn công mạng, cung cấp hiểu biết cơ bản về quy trình tấn công mạng.	Là một cơ sở kiến thức chuyên sâu tương quan với thông tin an ninh mạng theo hệ thống phân cấp Chiến thuật, Kỹ thuật, Quy trình và Kiến thức chung khác, chẳng hạn như phân bổ cho các nhóm đối thủ cụ thể



Cyber Kill Chain vs MITRE ATT&CK

Cyber Kill Chain

Không cung cấp những hiểu biết sâu sắc về các thủ tục của kẻ tấn công, làm hạn chế tính hữu dụng của nó.

Tuyên bố tất cả các cuộc tấn công mạng phải tuân theo một chuỗi chiến thuật tấn công cụ thể để đạt được thành công
Tập trung vào bảo mật vành đai
=> ngăn chặn một giai đoạn trong quy trình của kẻ tấn công sẽ vô hiệu hóa cuộc tấn công => không đủ

MITRE ATT&CK

ATT&CK đóng vai trò là danh sách kiểm tra các phương pháp và mục tiêu của kẻ tấn công, chứng minh việc đưa vào các biện pháp kiểm soát bảo mật và đảm bảo các biện pháp kiểm soát này là toàn diện cũng như cung cấp một số mức độ bảo vệ chống lại tất cả các khía cạnh của các cuộc tấn công mạng trong thế giới thực.

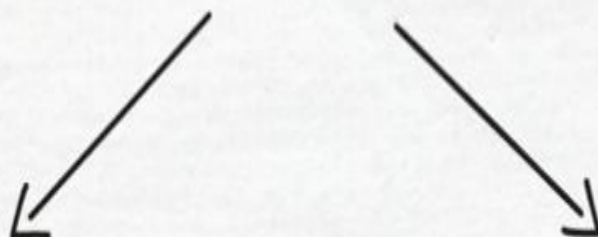
ATT&CK hữu ích hơn đối với những người săn tìm mối đe dọa, đội đỏ cũng như những người thiết kế và triển khai các chính sách và biện pháp kiểm soát bảo mật, chẳng hạn như kiến trúc sư và quản trị viên mạng và bảo mật.



2. THU THẬP THÔNG TIN

RECONNAISSANCE

INFORMATION GATHERING ABOUT THE TARGET



PASSIVE

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

ACTIVE

- NMAP
- PORT SCANNING
- BANNER GRABBING





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

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



2.1 Trích xuất thông tin từ Server với Shodan

- ❖ 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.



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



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



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



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

REST API Documentation

The base URL for all of these methods is:

<https://api.shodan.io>

Note: All API methods are rate-limited to 1 request/ 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



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]}
```



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



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))
```



Tìm kiếm cho FTP servers

```
8 #!/usr/bin/env python
9 import shodan
10 import re
11 import os
12 servers = []
13 SHODAN_API_KEY = os.environ['SHODAN_API_KEY']
14 shodanApi = shodan.Shodan(shodanKeyString)
15 results = shodanApi.search("port: 21 Anonymous user logged in")
16 print("hosts number: " + str(len( results['matches'])))
17 for result in results['matches']:
18     if result['ip_str'] is not None:
19         servers.append(result['ip_str'])
20 for server in servers:
21     print(server)
```

```
In [16]: runfile('D:/Google D
hosts number: 100
70.40.210.79
153.127.37.14
101.98.62.139
50.87.180.25
162.241.245.46
162.144.214.131
192.185.133.85
144.202.0.37
158.69.166.71
134.119.56.145
67.20.80.193
209.59.144.69
67.20.80.149
51.75.186.62
69.25.107.19
```



2.2 Bộ lọc Shodan và công cụ tìm kiếm BinaryEdge

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.



Tìm kiếm BinaryEdge

<https://www.binaryedge.io>

LOOK FOR SUBDOMAINS

Sub-domain enumeration. Discover hosts related to a specific domain.

Results for your query: *www.python.org*

75 results found.

Showing 1 to 75 of 75 entries.

Domains

chat.uk.python.org

empleo.es.python.org

dinsdale.python.org

pycon-archives.python.org

comunidad.es.python.org

Lấy tên miền phụ từ tên miền cụ thể python.org



Tìm kiếm BinaryEdge

```
$ sudo pip3 install pybinaryedge
```

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

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 FASTLY, US	334
80/tcp	142	Apache httpd	31	Germany	33	14061 DIGITALOCEAN-ASN, US	34
9999/tcp	4	nginx	29	France	28	63949 LINODE-AP Linode, LLC, US	33
5000/tcp	1	nginx/1.10.3 (Ubuntu)	17	United Kingdom	21	47570 V2O-SIA-AS, LV	18
8000/tcp	1	nginx/1.10.3	16	Latvia	18	20473 AS-CHOOPA, US	15

Thông tin tên miền cụ thể qua dịch vụ BinaryEdge



2.3. Sử dụng mô đun socket để lấy thông tin máy chủ

```
$ python3 get_banner_server.py -h
```

```
usage: get_banner_server.py [-h] -target -port 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'
```



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')`



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
```

```
...
```



2.5.Thu thập địa chỉ trên server với Fuzzing

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



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 https://github.com/attackercan/	4 years ago
regex	cross-updating with https://github.com/andresriancho/w3af/blob/master...	4 years ago
web-backdoors	Add files in asmx format	9 months ago
wordlists-misc	Resolvers file for subdomain brute force	2 years ago
wordlists-user-passwd	Update readme.txt	8 months ago
.gitignore	added Null representations for double encoding, format string %* 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

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



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)
```



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



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**



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',  
        'PortScannerYield',  
        'Process',  
        '__author__',  
        '__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__',
```



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



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 -O')
print(portScanner.command_line())
hosts_list = [(x, portScanner[x]['status']['state']) for x 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']))
```

```
In [4]: runfile('D:/Google Drive/ml_waf/untitled0.py', wdir='D:/Google Drive/ml_waf')

Host scan: 183.81.34.136
nmap -oX - -n -p21,22,23,25,80 183.81.34.136
183.81.34.136 up
Protocol : tcp
Port : 21 State : closed
Port : 22 State : closed
Port : 23 State : closed
Port : 25 State : closed
Port : 80 State : open
```

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



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.



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()
```

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

Chế độ đồng bộ



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)
```

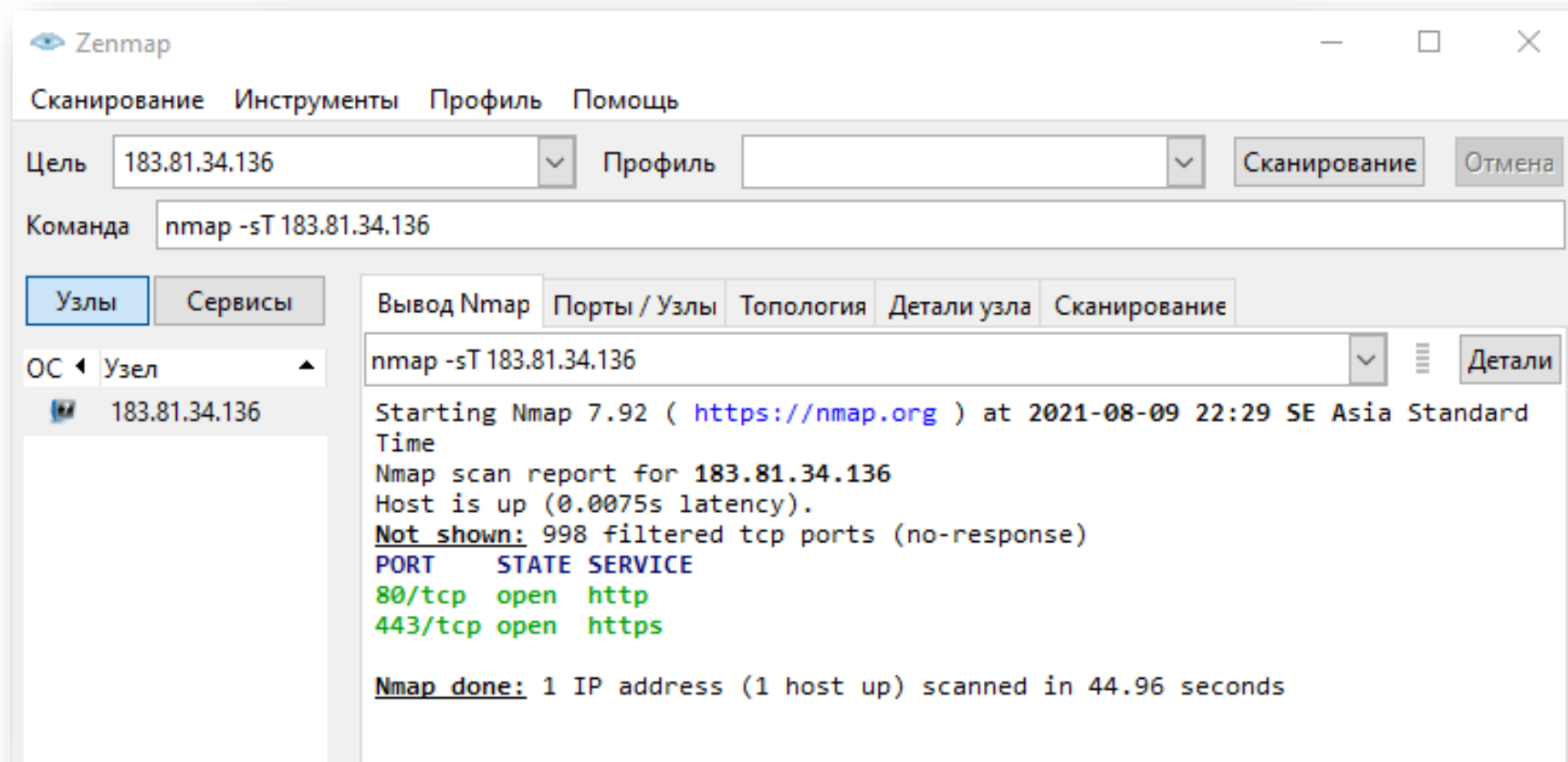
```
if self._process is not None:
```

Chế độ không đồng bộ



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)
```





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



Yêu cầu sinh viên chuẩn bị

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

- Nessus
- OpenVAS

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

- Acunetix
- CMSMap
- SQLmap

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

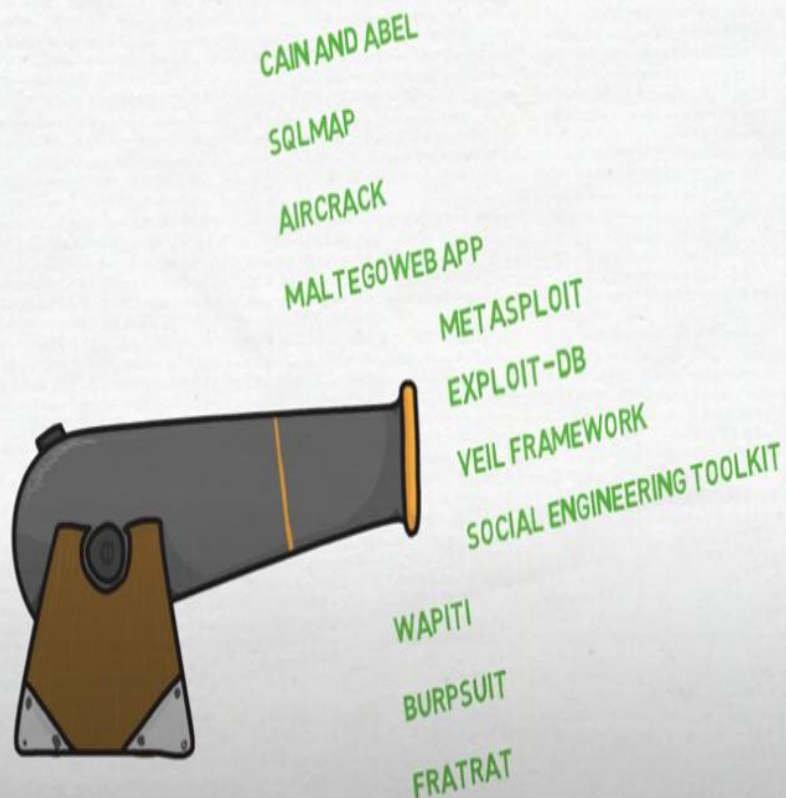
***Mỗi nhóm không quá 4 sinh viên**



3. CÔNG CỤ HÓA

WEAPONIZATION

FIND OR CREATE THE ATTACK TO EXPLOIT THE WEAKNESS



DEFENSIVE:

-PATCH MANAGEMENT
-DISABLE: OFFICE MACROS, JAVASCRIPT
AND BROWSER PLUGINS

SECURITY BASICS
-AV
-IPS
-EMAIL SECURITY
-MFA
-AUDIT LOGGING
-ECT..

ADMINISTRATIVE
CONTROLS:

TECHNICAL
CONTROLS:



Ví dụ về xây dựng payload cho tấn công xss

- `<SCRIPT>alert('XSS');</SCRIPT>`
`<script>alert('XSS');</script>`
`<BODY ONLOAD=alert('XSS')>`
`<SCR%00IPT>alert(\"'XSS'\")</SCR%00IPT>`
- ...



Identifying pages vulnerable to SQL injection

- ' or 'a'='a
' or 'x'='x
' or 0=0 #
' or 0=0 --
' or 1=1 or ''='
' or 1=1--
' ' or 1 --''

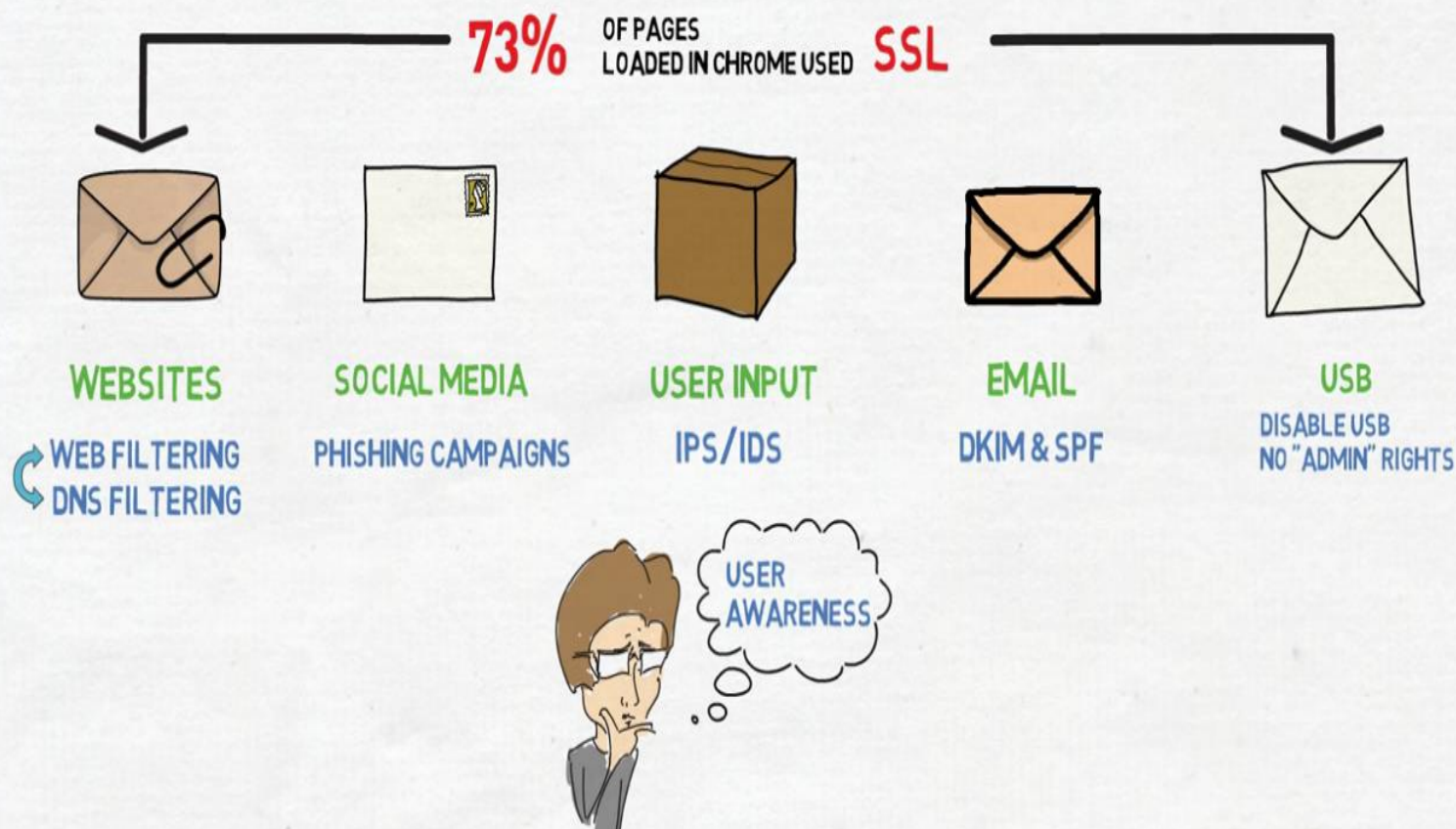
') or ('a'='a



4. Phân tán

DELIVERY

SELECTING WHICH AVENUE TO DELIVER THE EXPLOIT





Ẩn payload vào pixel ảnh .png

```
23  from PIL import Image
24
25  print("[*] Opening payload and converting to bit string")
26  payload = open(args.payload, 'r').read().encode('hex')
27  bin_payload = "".join('{:04b}'.format(int(c, 16)) for c in payload)
28
29  im = Image.open(args.inp).convert('RGBA')
30  pixels = im.load()
31  size = im.size[0]*im.size[1]
32
33  if len(bin_payload) > 3*size:
34      print("[*] Sorry, get a higher resolution image")
35      sys.exit()
36
37  def change_lsb():
38      index = 0
39      for j in range(0, im.size[1]):
40          for i in range(0, im.size[0]):
41              temp_list = list(pixels[i, j])
42              for k in [0, 1, 2]:
43                  temp_list[k] = ((pixels[i, j][k] & ~(1)) | (int(bin_payload[index])))
44                  index += 1
45              pixels[i, j] = tuple(temp_list)
46              if index == len(bin_payload): return
47
48  print("[*] Hiding data in LSB")
49  change_lsb()
50  print("[*] Saving intermediate PNG")
51  im.save("intermediate.png")
```




5. Khai thác



EXPLOITATION

WEAPON DELIVERED; ATTACK EXECUTED

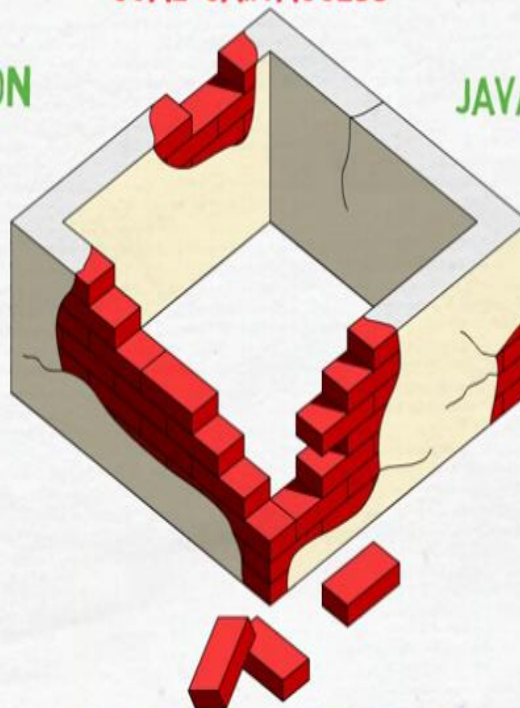
GOAL: GAIN ACCESS

SQL INJECTION

JAVASCRIPT HIJACK

BUFFER OVERFLOW

MALWARE



DATA EXECUTION PREVENTION (DEP)

ANTI-EXPLOIT



Buffer overflow

```
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] # Recieve IP from user
8 port = int(sys.argv[2]) # Recieve 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) # Recieve FTP Banner
16     client.send("USER." + "A" * length) # Send the user command with a variable length name
17     client.recv(1024) # Recieve Reply
18     client.send("PASS pass") # Send pass to complete connection attempt (will fail)
19     client.recv(1024) # Recieve 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
```



6. Cài đặt



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

-UBA/EDR

RESPOND

-FOLLOW INCIDENT RESPONSE SOPS
(I.D. DEVICE -> ISOLATE -> WIPE)

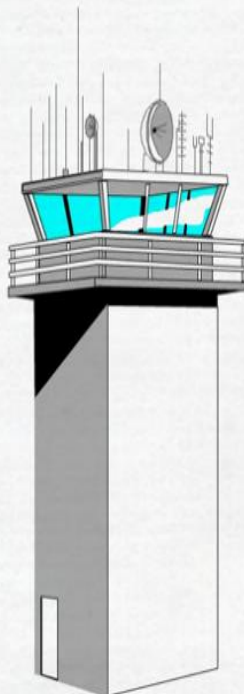
RECOVER

-RESTORE OR REIMAGE

GOAL: GAIN PERSISTANT ACCESS



7. CHỈ HUY VÀ KIỂM SOÁT



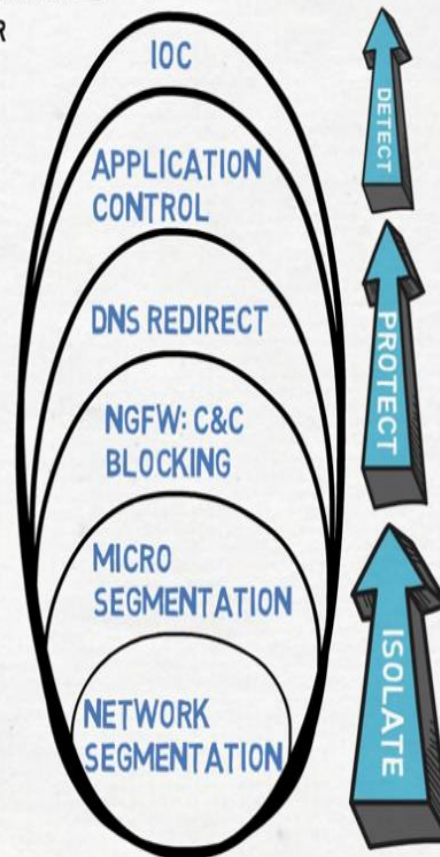
COMMAND AND CONTROL

REMOTE CONTROL OF THE SYSTEM BY THE ATTACKER

```
meterpreter > msv
[+] Running as SYSTEM
[*] Retrieving msv credentials
[*] msv credentials
```

AuthID	Package	Domain	User	Password
0:1035282	NTLM	WIN-LOANLTDQLU	Ralf	lm{ 00000000000000000000 }
0:1035232	NTLM	WIN-LOANLTDQLU	Ralf	lm{ 00000000000000000000 }
0:669397	NTLM	WIN-LOANLTDQLU	Fred	lm{ aad3b435b51404eeaad3b435b51404eeaad3b435b51404ee }
0:669366	NTLM	WIN-LOANLTDQLU	Fred	lm{ aad3b435b51404eeaad3b435b51404eeaad3b435b51404ee }
0:997	Negotiate	NT AUTHORITY	LOCAL SERVICE	n.s. (Credentials KO)
0:996	Negotiate	WORKGROUP	WIN-LOANLTDQLU\$	n.s. (Credentials KO)
0:42061	NTLM			n.s. (Credentials KO)
0:999	NTLM	WORKGROUP	WIN-LOANLTDQLU\$	n.s. (Credentials KO)

```
meterpreter >
```



☒ SSL DEEP PACKET INSPECTION



Phương pháp kết nối từ xa đến máy tính khác

- Socket
- WMI library
- Netuse mehod



VMI 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.")
```



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)
```



8. HÀNH ĐỘNG

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