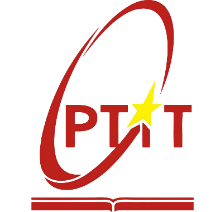
**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA AN TOÀN THÔNG TIN**

-------🙞🙜🕮🙞🙜-------



**BÁO CÁO BÀI THỰC HÀNH**

**Tìm hiểu về kỹ thuật T1057 trong Mitre Att&CK Process Discovery**

Sinh viên: Đỗ Mạnh Cường

Mã sinh viên: B20DCAT021

Giảng viên hướng dẫn: T.S Nguyễn Ngọc Điệp

**Hà Nội - 2025**

MỤC LỤC

[MỤC LỤC 1](#_Toc185618100)

[DANH MỤC CÁC HÌNH VẼ 2](#_Toc185618101)

[DANH MỤC CÁC BẢNG BIỂU 2](#_Toc185618102)

[DANH MỤC CÁC TỪ VIẾT TẮT 3](#_Toc185618103)

[1.1 Giới thiệu chung về bài thực hành 4](#_Toc185618104)

[1.2 Nội dung và hướng dẫn bài thực hành 4](#_Toc185618105)

[1.2.1 Mục đích 4](#_Toc185618106)

[1.2.2 Yêu cầu đối với sinh viên 4](#_Toc185618107)

[1.2.3 Nội dung thực hành 4](#_Toc185618108)

[1.3 Phân tích yêu cầu bài thực hành 6](#_Toc185618109)

[1.4 Thiết kế bài thực hành 6](#_Toc185618110)

[1.5 Cài đặt và cấu hình các máy ảo 9](#_Toc185618111)

[1.6 Tích hợp và triển khai 11](#_Toc185618112)

[1.6.1 Docker Hub 11](#_Toc185618113)

[1.6.2 Github 12](#_Toc185618114)

[1.7 Thử nghiệm và đánh giá 13](#_Toc185618115)

[TÀI LIỆU THAM KHẢO 16](#_Toc185618116)

DANH MỤC CÁC HÌNH VẼ

[Hình 1 Giao diện Labedit của bài lab 9](#_Toc185618117)

[Hình 2 Cài đặt phần Result 9](#_Toc185618118)

[Hình 3 Cài đặt phần Result (Continue) 9](#_Toc185618119)

[Hình 4 Cài đặt phần Goal 10](#_Toc185618120)

[Hình 5 Dockerfiles của máy client 10](#_Toc185618121)

[Hình 6 Dockerfiles của máy chủ server 11](#_Toc185618122)

[Hình 7 Add và commit bài lab 11](#_Toc185618123)

[Hình 8 Đẩy các vùng chứa lên dockerhub 12](#_Toc185618124)

[Hình 9 Tạo imodule.tar chứa bài thực hành 12](#_Toc185618125)

[Hình 10 Các vùng chứa được đẩy lên dockerhub 12](#_Toc185618126)

[Hình 11 Tạo file Imodule.tar 12](#_Toc185618127)

[Hình 12 File imodule.tar chứa bài thực hành 13](#_Toc185618128)

[Hình 13 Đẩy file imodule.tar lên github 13](#_Toc185618129)

[Hình 14 Quét dịch vụ TCP 13](#_Toc185618130)

[Hình 15 Quét dịch vụ UDP 14](#_Toc185618131)

[Hình 16 Brute force community string 14](#_Toc185618132)

[Hình 17 Discovery process. 14](#_Toc185618133)

[Hình 18 Đăng nhập auth server bằng credential mới lấy được. 15](#_Toc185618134)

[Hình 19 Capture the flag. 15](#_Toc185618135)

[Hình 20 Checkwork 16](#_Toc185618136)

DANH MỤC CÁC BẢNG BIỂU

[Bảng 1. Bảng Result 7](#_Toc183769440)

[Bảng 2. Bảng Goal 8](#_Toc183769441)

[Bảng 3. Bảng Parameter 8](#_Toc183769442)

DANH MỤC CÁC TỪ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| **Từ  viết tắt** | **Thuật ngữ tiếng Anh/Giải thích** | **Thuật ngữ tiếng Việt/Giải thích** |
| OID | Object Identifier | Mã nhận dạng đối tượng |
| MIB | Management Information Base | Cơ sở quản lý thông tin |
| OS | Operating System | Hệ điều hành |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

* 1. Giới thiệu chung về bài thực hành

Bài thực hành "Tìm hiểu về kỹ thuật Process Discovery T1057 trong Mitre ATT&CK Framework" được thiết kế nhằm giúp sinh viên nắm vững các khái niệm và kỹ thuật liên quan đến việc khám phá quy trình hệ thống như được mô tả trong khung Mitre ATT&CK. Đây là một bước quan trọng trong việc đào tạo kiến thức chuyên sâu về an ninh mạng, đặc biệt là cách thức các cuộc tấn công được thực hiện và phát hiện trong thực tế. Trong bài thực hành này, sinh viên sẽ tìm hiểu về kỹ thuật Process Discovery, được liệt kê trong ATT&CK Framework với mã kỹ thuật T1057, và các chiến thuật Discovery mà kẻ tấn công sử dụng để thu thập thông tin về môi trường hệ thống. Sinh viên sẽ học cách phân tích và mô phỏng các phương pháp mà kẻ tấn công có thể sử dụng để liệt kê các quy trình đang chạy, từ đó xác định các mục tiêu tiềm năng cho việc khai thác thêm. Qua đó, sinh viên sẽ nắm bắt được cách thức hoạt động của các quy trình hệ thống, các công cụ thường được sử dụng và các quyền cần thiết để thực hiện thành công cuộc tấn công.

Thông qua việc thực hiện bài lab, sinh viên không chỉ được trang bị kiến thức về cách phát hiện và ngăn chặn các hoạt động Process Discovery mà còn nâng cao kỹ năng phân tích và bảo vệ các hệ thống mạng trước các mối đe dọa tiềm ẩn. Điều này sẽ giúp sinh viên phát triển nền tảng vững chắc về bảo mật thông tin và ứng dụng thực tế trong việc bảo vệ an ninh hệ thống.

* 1. Nội dung và hướng dẫn bài thực hành
     1. Mục đích

Giúp sinh viên hiểu rõ cách mà kẻ tấn công sử dụng để liệt kê các quy trình đang chạy trong hệ thống từ đó nâng cao khả năng nhận diện và đánh giá các hoạt động bất thường trên hệ thống.

* + 1. Yêu cầu đối với sinh viên

Có kiến thức cơ bản về hệ điều hành Linux, hiểu cơ chế hoạt động của giao thức SNMP và cách thức, quy trình thu thập thông tin về đối tượng.

* + 1. Nội dung thực hành

Khởi động bài lab:

Vào terminal, gõ:

*labtainer -r pen\_bsic\_recon\_t1057*

*(chú ý: sinh viên sử dụng mã sinh viên của mình để nhập thông tin email người thực hiện bài lab khi có yêu cầu, để sử dụng khi chấm điểm)*

Sau khi khởi động xong một terminal ảo sẽ xuất hiện, đại diện cho máy tấn công: ***client***.

Sử dụng công cụ nmap để quét các dịch vụ đang chạy trên server. VD:

*nmap -sT -p- --min-rate 5000 -sV -sC -oA nmap/alltcp 172.50.2.20*

Quét cả những cổng UDP với options -sU

Kết quả rà quét cho thấy server đang sử dụng dịch vụ SNMP. Khai thác lỗ hổng misconfig community string bằng cách brute-force với công cụ onesixtyone và public wordlist. (SecLists, DiscoveryList, …)

*onesixtyone -c dict.txt -i hosts -o my.log -w 100*

Sau khi có được community string của SNMP agent. Sử dụng các công cụ như snmpwalk, snmpget, … cùng với các OID trong MIB để rà quét thông tin về server như thông tin giao diện mạng (IPv4, IPv6), username, uptimes, OS version, process running,…

*snmpwalk -v [VERSION\_SNMP] -c [COMM\_STRING] [DIR\_IP] .1*

Sử dụng những OID truy vấn thông tin về các process đang chạy như: .1.3.6.1.2.1.25.4.2,…

Tìm kiếm những thông tin nhạy cảm hữu ích từ process list.

Trên port 8443 đang chạy một dịch vụ HTTP Auth Server, … (dùng curl). Nhập những thông tin đăng nhập bị lộ lọt để truy cập vào trang quản lý thư mục. Đọc file flag.txt

*curl -v 172.50.2.20:8443/flag.txt -u username:password*

Kết thúc bài lab:

Trên terminal đầu tiên sử dụng câu lệnh sau để kết thúc bài lab:

*stoplab pen\_bsic\_recon\_t1057*

Khi bài lab kết thúc, một tệp zip lưu kết quả được tạo và lưu vào một vị trí được hiển thị bên dưới stoplab.

Khởi động lại bài lab:

Trong quá trình làm bài sinh viên cần thực hiện lại bài lab, dùng câu lệnh:

*labtainer –r pen\_bsic\_recon\_t1057*

* 1. Phân tích yêu cầu bài thực hành

Bài thực hành gồm 2 máy tính nằm trên 1 dải mạng. Trong đó một máy là client và một máy là server đang chạy SNMP service để quản lý các thiết bị mạng. Để hoàn thành bài thực hành sinh viên cần thu thập thông tin về máy chủ SNMP lợi dụng lỗ hổng trong cấu hình dịch vụ.

* 1. Thiết kế bài thực hành

Trên môi trường máy ảo Ubuntu được cung cấp, sử dụng docker tạo ra 2 container: 1 container mang tên “client” đóng vai trò máy khách, 1 container mang tên “server” đóng vai trò máy chủ.

Tạo 1 dải mạng có cấu hình: 172.50.2.0/24 và gateway: 172.50.2.1.

Cấu hình docker gồm có:

Client: lưu cấu hình cho máy khách, trong đó gồm có:

Tên máy: client

Địa chỉ trong mạng: 172.50.2.10

Server: lưu cấu hình cho máy chủ, trong đó gồm có:

Tên máy: server

Địa chỉ trong mạng: 172.50.2.20

config: lưu cấu hình hoạt động của hệ thống

dockerfiles: mô tả cấu hình của container trong đó:

Server chạy dịch vụ SNMP để quản lý các thiết bị mạng và một dịch vụ Web trên cổng 8443 dùng để xác thực user truy cập đến tài nguyên trên máy chủ.

Client chạy như một thiết bị mạng được quản lý bởi server.

Các nhiệm vụ cần phải thực hiện để thực hành thành công:

Sinh viên cần xác định các dịch vụ TCP/UDP đang chạy trên server bằng nmap.

Khai thác lỗ hổng trong cấu hình sai SNMP service.

Brute-force community string bằng công cụ onesixtyone,… và common community string.

Với community string tìm được, dùng nó để liệt kê các thông tin về OS, process, … trên server sử dụng bộ công cụ của snmp như snmpwalk, snmpget, …

Tìm kiếm trong các thông tin thu thập được những dữ liệu nhạy cảm như credential,…

Dùng thông tin tìm kiếm được để truy cập tới Auth Server, cho phép truy vấn các thư mục trên máy chủ. Từ đó lấy flag từ file /flag.txt

Kết thúc bài lab và đóng gói kết quả.

Thiết lập hệ thống mạng sao cho máy attacker và máy client cùng một mạng LAN.

Để đánh giá được sinh viên đã hoàn thành bài thực hành hay chưa, cần chia bài thực hành thành các nhiệm vụ nhỏ, mỗi nhiệm vụ cần phải chỉ rõ kết quả để có thể dựa vào đó đánh giá, chấm điểm. Do vậy, trong bài thực hành này hệ thống cần ghi nhận các thao tác, sự kiện được mô tả và cấu hình như bảng 1,2,3:

1. Bảng Result

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Result Tag | Container | File | Field Type | Field ID | Timestamp Type | LINE ID |
| -scan\_service\_snmp | client | nmap.stdout | CONTAINS | Snmp | FILE |  |
| \_scan\_udp | client | nmap.stdout | CONTAINS | 161 | FILE |  |
| \_scan\_tcp | client | nmap.stdout | CONTAINS | 8443/tcp | FILE |  |
| \_scan\_service\_http | client | nmap.stdout | CONTAINS | http | FILE |  |
| brute\_force | client | onesixtyone.stdout | CONTAINS | pr1v4t3 | FILE |  |
| enum\_info | client | \*.stdout | CONTAINS | --username admin | FILE |  |
| \_get\_credential1 | client | curl.stdout | CONTAINS | flag.txt | FILE |  |
| \_get\_credential2 | client | curl.stdout | CONTAINS | HTTP/1.0 200 OK | FILE |  |
| \_get\_flag\_result | client | curl.stdout | TOKEN | LAST | HAVESTRING | flag |

- brute\_force: Dùng công cụ onesixtyone và một tập wordlist phổ biến để brute-force thành công snmp community string.

- enum\_info: Dùng công cụ SNMPWALK để thu thập, liệt kê các thông tin về OS, process info, … từ đó lấy được thông tin đăng nhập vào auth server.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Goal ID | Goal Type | Operator | Result Tag | Answer Type | Parameter |
| scan\_tcp | boolean | \_scan\_tcp and \_scan\_service\_http |  |  |  |
| scan\_udp | boolean | \_scan\_udp and \_scan\_service\_snmp |  |  |  |
| curl\_auth\_server | boolean | \_get\_credential1 and \_get\_credential2 |  |  |  |
| get\_flag | matchany | string\_contains | \_get\_flag\_result | Parameter | flag\_ |

1. Bảng Goal

* scan\_tcp: Sử dựng nmap quét và xác định được sự tồn tại của một port đang chạy dịch vụ HTTP.
* scan\_udp: Sử dụng nmap để quét và xác định được sự tồn tại của một port đang chạy dịch vụ SNMP
* curl\_auth\_server: Truy cập thành công tới HTTP Auth Server cùng với thông tin đăng nhập đã lấy được bằng curl.
* get\_flag: Đọc được file /flag.txt

1. Bảng Parameter

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Param ID | Operator | File name | Symbol | Step | Lower bound | Upper bound |
| flag\_ | RAND\_REPLACE\_UNIQUE | server:/flag.txt | FLAG | 1 | 215987415 | 915987415 |

* Với mỗi sinh viên thực hiện việc đọc file flag.txt, flag sẽ được thay đổi ngẫu nhiên đảm bảo cho việc cá nhân hóa dữ liệu.
  1. Cài đặt và cấu hình các máy ảo

A screenshot of a computer

Description automatically generated

1. Giao diện Labedit của bài lab

A screenshot of a computer

Description automatically generated

1. Cài đặt phần Result

A screenshot of a computer

Description automatically generated

1. Cài đặt phần Result (Continue)

A screenshot of a computer

Description automatically generated

1. Cài đặt phần Goal

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

1. Dockerfiles của máy client

A screenshot of a computer

Description automatically generated

1. Dockerfiles của máy chủ server
   1. Tích hợp và triển khai

Bài thực hành đã được triển khai như sau:

Docker Hub

https://hub.docker.com/u/domanhcuong2502

A screenshot of a computer program

Description automatically generated

1. Add và commit bài lab

A computer screen shot of a program

Description automatically generated

1. Đẩy các vùng chứa lên dockerhub

A computer screen shot

Description automatically generated

1. Tạo imodule.tar chứa bài thực hành

A screenshot of a computer

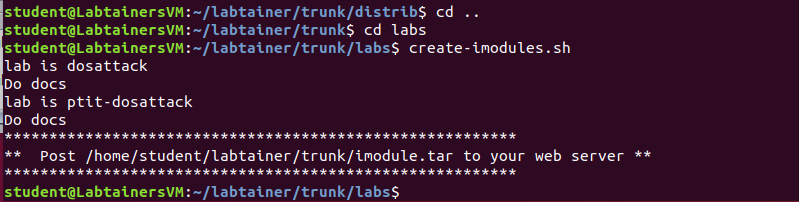
Description automatically generated

1. Các vùng chứa được đẩy lên dockerhub

Github

https://github.com/oceanmancuonh/pen\_bsic\_recon\_t1057

Nhập lệnh create-imodules.sh



1. Tạo file Imodule.tar

A group of orange folders

Description automatically generated

1. File imodule.tar chứa bài thực hành

Tạo repo mới để đẩy imodule.tar lên và tạo phần release mới

A screenshot of a computer

Description automatically generated

1. Đẩy file imodule.tar lên github
   1. Thử nghiệm và đánh giá

Bài thực hành đã được xây dựng thành công, dưới đây là hình ảnh minh họa về bài thực hành:

A computer screen with white text

Description automatically generated

1. Quét dịch vụ TCP

A computer screen with white text

Description automatically generated

1. Quét dịch vụ UDP

A computer screen with white text

Description automatically generated

1. Brute force community string

A computer screen with white text

Description automatically generated

1. Discovery process.

A screenshot of a computer program

Description automatically generated

1. Đăng nhập auth server bằng credential mới lấy được.

A computer screen shot of a program

Description automatically generated

1. Capture the flag.

A computer screen shot of a program

Description automatically generated

1. Checkwork

TÀI LIỆU THAM KHẢO

[1] Polop, C. (n.d.). *161,162,10161,10162/udp - Pentesting SNMP*. Retrieved from https://hacktricks.boitatech.com.br/pentesting/pentesting-snmp

[2] Philip Pickering, & Gorjan Petrovski. (n.d.). *Script snmp-brute.* Retrieved from https://nmap.org/nsedoc/scripts/snmp-brute.html

[3] trickster0. (n.d.). *Mischief - HackTheBox*. Retrieved from https://www.hackthebox.com/machines/mischief