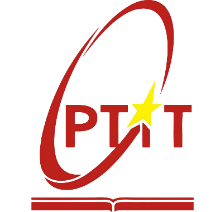
**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA AN TOÀN THÔNG TIN**

-------🙞🙜🕮🙞🙜-------



**BÁO CÁO BÀI THỰC HÀNH**

**Tìm hiểu về kỹ thuật T1184 trong Mitre Att&CK SSH Hijacking**

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](#_Toc185938240)

[DANH MỤC CÁC HÌNH VẼ 2](#_Toc185938241)

[DANH MỤC CÁC BẢNG BIỂU 2](#_Toc185938242)

[DANH MỤC CÁC TỪ VIẾT TẮT 3](#_Toc185938243)

[1.1 Giới thiệu chung về bài thực hành 4](#_Toc185938244)

[1.2 Nội dung và hướng dẫn bài thực hành 4](#_Toc185938245)

[1.2.1 Mục đích 4](#_Toc185938246)

[1.2.2 Yêu cầu đối với sinh viên 4](#_Toc185938247)

[1.2.3 Nội dung thực hành 4](#_Toc185938248)

[1.3 Phân tích yêu cầu bài thực hành 6](#_Toc185938249)

[1.4 Thiết kế bài thực hành 6](#_Toc185938250)

[1.5 Cài đặt và cấu hình các máy ảo 9](#_Toc185938251)

[1.6 Tích hợp và triển khai 12](#_Toc185938252)

[**1.6.1** Docker Hub 12](#_Toc185938253)

[**1.6.2** Github 13](#_Toc185938254)

[1.7 Thử nghiệm và đánh giá 14](#_Toc185938255)

[TÀI LIỆU THAM KHẢO 18](#_Toc185938256)

DANH MỤC CÁC HÌNH VẼ

[Hình 1 Mô hình mạng 7](#_Toc185938209)

[Hình 2 Giao diện Labedit của bài lab 9](#_Toc185938210)

[Hình 3 Cài đặt phần Result 10](#_Toc185938211)

[Hình 4 Cài đặt phần result 10](#_Toc185938212)

[Hình 5 Cài đặt phần Goal 10](#_Toc185938213)

[Hình 6 Dockerfiles của máy attacker 11](#_Toc185938214)

[Hình 7 Dockerfiles của client 11](#_Toc185938215)

[Hình 8 Dockerfile của jumpbox 12](#_Toc185938216)

[Hình 9 Dockerfile của server 12](#_Toc185938217)

[Hình 10 Add và commit bài lab 12](#_Toc185938218)

[Hình 11 Đẩy các vùng chứa lên dockerhub 13](#_Toc185938219)

[Hình 12 Tạo imodule.tar chứa bài thực hành 13](#_Toc185938220)

[Hình 13 Tạo file Imodule.tar 14](#_Toc185938221)

[Hình 14 Đẩy file imodule.tar lên github 14](#_Toc185938222)

[Hình 15 Tạo một cặp khóa SSH mới 14](#_Toc185938223)

[Hình 16 Chuyển public key tới máy jumpbox 15](#_Toc185938224)

[Hình 17 SSH tới máy jumpbox thay vì nhập password thì nhập passphrase 15](#_Toc185938225)

[Hình 18 Truyền public key tới server. 15](#_Toc185938226)

[Hình 19 Trở về client session, start một ssh agent. 15](#_Toc185938227)

[Hình 20 Thêm cặp key vào để ssh-agent quản lý. 16](#_Toc185938228)

[Hình 21 Tạo entry mới trong ssh config để forward ssh agent tới máy jumpbox. 16](#_Toc185938229)

[Hình 22 SSH thẳng từ client tới server mà không cần nhập password, passphrase. 16](#_Toc185938230)

[Hình 23 Attack ssh vào máy jumpbox với giả thuyết đã có được thông tin đăng nhập. 16](#_Toc185938231)

[Hình 24 Truy vấn các tiến trình đang chạy của user ubuntu. 17](#_Toc185938232)

[Hình 25 Hijacking ssh socket. 17](#_Toc185938233)

[Hình 26 SSH thành công tới target server mà không cần biết pw, passphrase. 17](#_Toc185938234)

[Hình 27 Checkwork. 17](#_Toc185938235)

DANH MỤC CÁC BẢNG BIỂU

[Bảng 1. Bảng Result 8](#_Toc185938238)

[Bảng 2. Bảng Goal 9](#_Toc185938239)

DANH MỤC CÁC TỪ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| **Từ  viết tắt** | **Thuật ngữ tiếng Anh/Giải thích** | **Thuật ngữ tiếng Việt/Giải thích** |
| PID | Process Identifying Digits | Nhận dạng chữ số xử lý |
| LAN | Local Area Network | Mạng cục bộ |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

* 1. Giới thiệu chung về bài thực hành

Bài thực hành "Tìm hiểu về kỹ thuật khai thác SSH Hijacking trong Mitre ATT&CK Framework" được thiết kế nhằm giúp sinh viên nắm vững các khái niệm và kỹ thuật liên quan đến chiếm đoạt phiên SSH 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 SSH Hijacking, được liệt kê trong ATT&CK Framework với mã kỹ thuật T1563.001, và các chiến thuật Lateral Movement mà kẻ tấn công sử dụng để di chuyển bên trong hệ thống mạng. Sinh viên sẽ học cách phân tích và mô phỏng các phương pháp mà kẻ tấn công có thể sử dụng để chiếm đoạt một phiên SSH hợp pháp, từ đó kiểm soát hệ thống mục tiêu. Qua đó sinh viên sẽ nắm bắt được cách thức hoạt động của các kết nối SSH, các yêu cầu hệ thống và 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 cuộc tấn công SSH Hijacking 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 tìm hiểu cách attacker khai thác lỗ hổng SSH Hijacking (tactic Lateral Moverment trong Mitre ATT&CK Framework) để đánh cắp SSH session, từ đó di chuyển ngang tới các tài nguyên khác trên server.

* + 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 SSH.

* + 1. Nội dung thực hành

Khởi động bài lab:

Vào terminal, gõ:

*labtainer -r pen\_bsic\_lateral\_mov\_t1184*

*(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 hai terminal ảo sẽ xuất hiện, một cái là đại diện cho máy tấn công: ***attacker***, một cái là đại diện cho máy khách: ***client***.

Trước hết sinh viên đóng vai trò như một client. Trên terminal ***client*** thực hiện thực hiện tạo một cặp khóa ssh bằng câu lệnh:

*ssh-keygen -t rsa*

Nhập một passphrase (Không để trống). Sau đó copy public key tới file *authorized\_keys* trên máy jumpbox. Điều này cho phép máy jumpbox chấp nhận bằng chứng sở hữu khóa riêng tương ứng làm cơ sở để xác thực client. (Password của user ubuntu trên jumpbox là: *ubuntu*)

*ssh-copy-id -i ~/.ssh/id\_rsa.pub jumpbox*

Bây giờ chúng ta có thể ssh từ client tới jumpbox mà không cần nhập password mà cần nhập passphrase.

Trong ssh session ở máy jumpbox tiếp tục copy public key ở file *authorized\_keys* tới target server. (Password của user ubuntu trên server là: *ubuntu)*

*scp -r ~/.ssh server:~/*

Tiếp theo trở về máy client. Sử dụng command sau để mở một SSH agent trên client và cung cấp cho agent quyền truy cập vào khóa riêng mới tạo của bạn.

*eval `ssh-agent`*

*ssh-add*

Khi đó ta có thể ssh vào jumpbox mà không cần nhập password hay passphrase. Nhưng khi ssh vào server thì vẫn cần password vì jumpbox không thể chứng minh rằng user yêu cầu sở hữu khóa riêng cho public key. Do đó ta sẽ bật tính năng cho phép chuyển tiếp kết nối SSH agent bằng cách cấu hình ssh.

Để cấu hình SSH agent forwarding ta có thể làm bằng cách thêm một entry vào file ***~/.ssh/config***có dạng:

*Host server*

*HostName server*

*ForwardAgent yes*

Giờ đây ta có thể ssh thẳng từ client tới server mà không cần nhập mật khẩu hay passphrase.

Khi đó sẽ có một tiến trình con duy trì SSH agent chạy dưới tiến trình ssh trong máy jumpbox. Sinh viên đóng vai trò như một attacker. Trong session terminal của attacker ssh tới máy jumpbox. (Giả sử ở đây attacker đã lấy được thông tin đăng nhập của user ubuntu trong jumpbox). Truy vấn processes của user khác.

Ex: *ps aux ww* or *pstree -p <user>*

Lưu ý processes id (PID) của shell được sử dụng bởi sshd. Ex: bash, sh,…

Tìm biến môi trường sử dụng trong PID đó trong file /proc/{PID}/environ (sử dụng grep để tìm *SSH\_AUTH\_SOCK*)

Sau khi tìm được vị trí, sử dụng socket đó để pivot đến server như client. Khai thác thành công

Kết thúc bài lab:

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

*stoplab* *pen\_bsic\_lateral\_mov\_t1184*

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\_lateral\_mov\_t1184*

* 1. Phân tích yêu cầu bài thực hành

Bài thực hành gồm 4 máy tính nằm trên 2 dải mạng. Trong đó có 2 máy được coi như là client, 1 máy được coi là máy chủ trung gian và cuối cùng là máy chủ mục tiêu. Để hoàn thành bài thực hành sinh viên cần lấy được SSH session của client đang được lưu trữ thông qua SSH agent đang sử dụng để có thể truy cập trái phép vào máy chủ nội bộ mà không cần biết mật khẩu.

* 1. Thiết kế bài thực hành

Trên môi trường máy ảo Ubuntu được cung cấp, sử dụng docker tạo ra 4 container: 1 container mang tên “attacker” đóng vai trò máy tấn công, 1 container mang tên “client” đóng vai trò máy khách, 1 container mang tên “jumpbox” đóng vai trò là một máy chủ trung gian và 1 container mang tên “server” (target-host) đóng vai trò là máy chủ nội bộ.

A graphic of a phone jumpbox

Description automatically generated

1. Mô hình mạng

Tạo 1 dải mạng LOCAL\_LAN có cấu hình: 172.20.0.0/24 và gateway: 172.20.0.1 và một dải mạng REMOTE\_LAN có cấu hình: 172.30.0.0/24 với gateway: 172.30.0.1.

Cấu hình docker gồm có:

Attacker: lưu cấu hình cho máy tấn công, trong đó gồm có:

Tên máy: attacker

Địa chỉ trong mạng LOCAL\_LAN: 172.20.0.5

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

Tên máy: client

Địa chỉ trong mạng LOCAL\_LAN: 172.20.0.2

Jumpbox: lưu cấu hình máy chủ trung gian, trong đó gồm có 2 card mạng:

Tên máy: jumpbox

Địa chỉ trong mạng LOCAL\_LAN: 172.20.0.3

Địa chỉ trong mạng REMOTE\_LAN: 172.30.0.3

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

Tên máy: server

Địa chỉ trong mạng REMOTE\_LAN: 172.30.0.5

config: lưu cấu hình hoạt động của hệ thống

dockerfiles: mô tả cấu hình của container trong đó:

4 container đều sử dụng ubuntu image và enable ssh service.

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

Sinh viên đóng vai trò là một client, tạo một cặp key mới trên máy client. Chuyển cặp key đấy tới máy jumpbox và máy server

Bật SSH agent và cấu hình AgentForwarding trên máy client. SSH thành công tới server từ client mà cần không nhập mật khẩu hay passphrase.

Sinh viên đóng vai trò là attacker, attack được cung cấp credentials của máy jumpbox. Từ máy attacker ssh tới máy jumpbox. Truy vấn PID của ssh-agent process đang chạy. Từ đó lấy được shell variable SSH\_AUTH\_SOCK.

Sử dụng giá trị socket này (vốn lưu private key của client) để ssh tới server với session của client mà không cần biết password của user hay passphrase của key.

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 |
| \_welcome | client | ssh.stdout | CONTAINS | Welcome to server @172.30.0.5 | FILES |  |
| \_password\_prompt | client | ssh.stdout | CONTAINS | 's password | FILES |  |
| \_passphrase\_prompt | client | ssh.stdout | CONTAINS | Enter passphrase for key | FILES |  |
| ssh\_socket | attacker | \*.stdout | CONTAINS | SSH\_AUTH\_SOCK= | FILES |  |
| successful\_attack | attacker | ssh.stdout | CONTAINS | Welcome to server @172.30.0.5 | FILES |  |

- ssh\_socket: Truy vấn thành công SSH Socket đang chạy trên máy jumpbox

- successful\_attack: Thành công ssh tới target server từ máy attacker

|  |  |  |
| --- | --- | --- |
| Goal ID | Goal Type | Boolean |
| create\_agent | boolean | \_welcome and\_not (\_password\_prompt or \_passphrase\_prompt) |

1. Bảng Goal

* create\_agent: Tạo thành công Forwarding Agent rồi SSH thẳng từ client tới server mà không cần mật khẩu hay passphrase.
  1. Cài đặt và cấu hình các máy ảo

A screenshot of a computer

Description automatically generated

1. Giao diện Labedit của bài lab

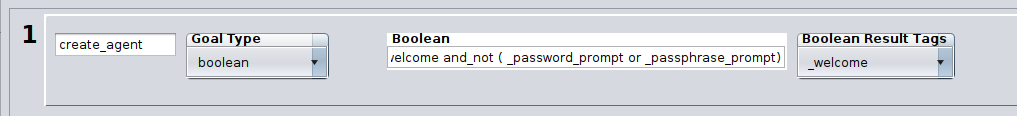


1. Cài đặt phần Result

A screenshot of a computer

Description automatically generated

1. Cài đặt phần result



1. Cài đặt phần Goal

A screenshot of a computer

Description automatically generated

1. Dockerfiles của máy attacker

A screenshot of a computer

Description automatically generated

1. Dockerfiles của client

A screenshot of a computer

Description automatically generated

1. Dockerfile của jumpbox

A screenshot of a computer

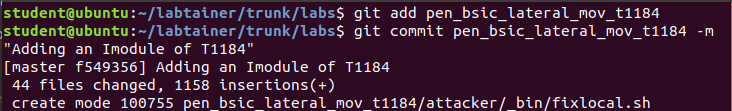
Description automatically generated

1. Dockerfile của server
   1. Tích hợp và triển khai

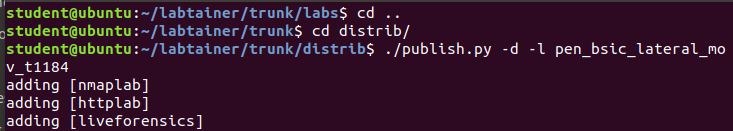
Bài thực hành đã được triển khai như sau:

Docker Hub

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



1. Add và commit bài lab



1. Đẩy các vùng chứa lên dockerhub

A screenshot of a computer screen

Description automatically generated

1. Tạo imodule.tar chứa bài thực hành

A screenshot of a computer

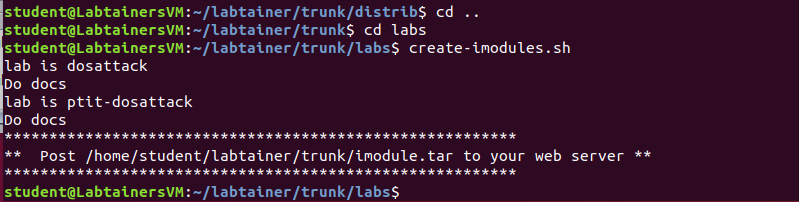
Description automatically generated

Hình 10. Các vùng chứa được đẩy lên dockerhub

Github

https://github.com/oceanmancuonh/pen\_bsic\_lateral\_mov\_t1184

Nhập lệnh create-imodules.sh



1. Tạo file Imodule.tar

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

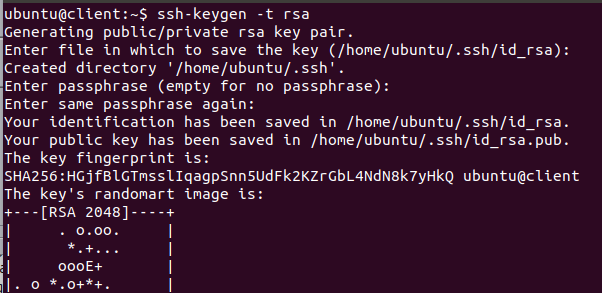
A screenshot of a computer

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:

Đóng vai trò là một client.



1. Tạo một cặp khóa SSH mới

A computer screen with white and blue text

Description automatically generated

1. Chuyển public key tới máy jumpbox

A computer screen with white text

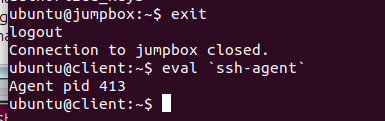
Description automatically generated

1. SSH tới máy jumpbox thay vì nhập password thì nhập passphrase

A computer screen with white text

Description automatically generated

1. Truyền public key tới server.



1. Trở về client session, start một ssh agent.

A computer screen shot of white text

Description automatically generated

1. Thêm cặp key vào để ssh-agent quản lý.

A screenshot of a computer

Description automatically generated

1. Tạo entry mới trong ssh config để forward ssh agent tới máy jumpbox.

A computer screen shot of a computer

Description automatically generated

1. SSH thẳng từ client tới server mà không cần nhập password, passphrase.

Đóng vai trò là attacker.

A computer screen with white text

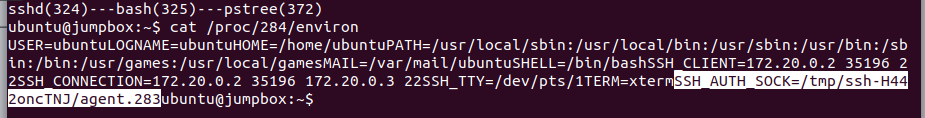
Description automatically generated

1. Attack ssh vào máy jumpbox với giả thuyết đã có được thông tin đăng nhập.

A screenshot of a computer

Description automatically generated

1. Truy vấn các tiến trình đang chạy của user ubuntu.



1. Hijacking ssh socket.

A computer screen with white text

Description automatically generated

1. SSH thành công tới target server mà không cần biết pw, passphrase.

A screenshot of a computer

Description automatically generated

1. Checkwork.

TÀI LIỆU THAM KHẢO

[1] AttackDefense. (n.d.). *T1184: SSH Hijacking*. Retrieved from https://attackdefense.com/challengedetails?cid=1764

[2] Clockwork. (n.d.). *SSH Agent Hijacking.* Retrieved from https://www.clockwork.com/insights/ssh-agent-hijacking

[3] wunderwuzzi. (n.d.). Retrieved from TTP Diaries: SSH Agent Hijacking: https://embracethered.com/blog/posts/2022/ttp-diaries-ssh-agent-hijacking/