****T1184: SSH Hijacking****

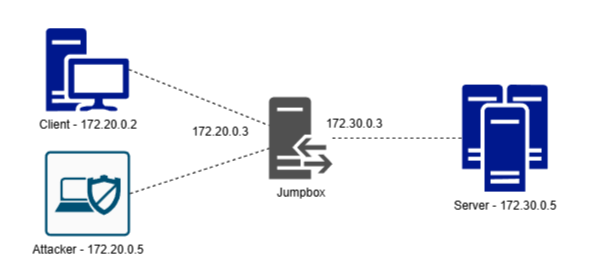
1. **Mục đích**

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, cơ chế hoạt động của SSH agent forwarding.

1. **Nội dung thực hành**

**

*Mô hình mạng bài thực hành*

Bài thực hành được thiết kế gồm:

- 2 dải mạng:

+ Mạng LOCAL\_LAN: 172.20.0.0/24 dành cho các máy nội bộ, gồm các máy

* Client: 172.20.0.2, đóng vai trò máy khách
* Attacker: 172.20.0.5, đóng vai trò là kẻ tấn công
* Jumpbox: 172.20.0.3, đóng vai trò máy chủ trung gian

+ Mạng REMOTE\_LAN: 172.30.0.0/24, gồm các máy:

* Jumpbox: 172.30.0.3, đóng vai trò máy chủ trung gian
* Server: 172.30.0.5 đóng vai trò máy chủ nội bộ

**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*

**Lưu ý:** server ở đây là nơi khi ssh vào thì agent trên client sẽ được chuyển tiếp vào phiên ssh đó.  
Khi cấu hình xong, ta có thể ssh thằng từ client đến jumpbox, rồi ssh từ jumpbox đến server mà không cần nhập mật khẩu hay passphrase vì khi đó quá trình xác thực sẽ sử dụng agent được forward từ client, chứ không cần khóa riêng lưu tại jumpbox.

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>*

Tiến trình kiểu như: sshd → bash → .... Ghi nhớ **PID của shell**. 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)

*cat /proc/{PID}/environ*

Sau khi tìm được biến môi trường , sử dụng socket đó để pivot đến server như client. Đặt lại biến môi trường đã tìm được để dùng socker đã tìm được và thực hiện ssh vào server

*export SSH\_AUTH\_SOCK={…}*

*ssh ubuntu@server*

Nếu không cần nhập mật khẩu hay passphrase thì ta đã khai thác SSH Hijacking thành công.

**Kiểm tra kết quả:**

Sinh viên kiểm tra tiến độ hoặc kết quả bài thực hành với câu lệnh:

*checkwork*

**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*