HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
KHOA AN TOÀN THÔNG TIN

Báo cáo bài thực hành

Phát hiện và phản ứng sự cố Port scanning

Sinh viên thực hiện:

B20DCAT002 – Hoàng Thu Cúc

Giảng viên hướng dẫn: TS.Nguyễn Ngọc Điệp

HÀ NỘI 12-2025

MỤC LỤC

DANH MỤC CÁC HÌNH VẼ

NỘI DUNG THỰC HÀNH

1.1 Lab 4: Phát hiện và phản ứng sự cố Port scanning

1.1.1 Giới thiệu chung

Bài thực hành "Phát hiện và phản ứng sự cố tấn công Port Scanning" giúp sinh viên hiểu về :

- quy trình thu thập log mạng, nhận diện hành vi quét cổng trái phép
- cấu hình Splunk để thu thập log TCP SYN từ client
- cách triển khai iptables để chặn nguồn tấn công

1.1.2 Mục đích

Sau khi hoàn thành bài thực hành, sinh viên có thể:

- Hiểu hơn về cơ chế port scanning, vai trò của gói TCP SYN và các dấu hiệu bất thường trong hoạt động quét cổng.
- Dùng tcpdump và Splunk để thu thập và phân tích log quét cổng.
- Dùng iptables để chặn nguồn tấn công

1.1.3 Yêu cầu đối với sinh viên

- Có kiến thức cơ bản về Linux, tcpdump và iptables.
- Hiểu khái niệm port scanning, SYN scan và đăng tải log mạng vào Splunk.
- Nắm được nguyên lý thu thập log bằng Splunk Universal Forwarder và cách phân tích log cơ bản trong Splunk.

1.1.4 Nội dung thực hành

- Trước khi khởi động bài lab, cần đảm bảo labtainer được cấu hình như sau:
    - Memory (RAM): 10GB
    - Hard Disk: Tối thiểu 80GB (khuyến nghị 100GB)
- Tải bài lab:

Vào /home/student/labtainer/labtainer-student và gõ lệnh sau trên terminal:

imodule https://github.com/anhdnmit/do_an_tot_nghiep/raw/refs/heads/main/idr_splunk_portscanning/imodule.tar

- Khởi động bài lab:

labtainer -r idr_splunk_portscanning

(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, 3 terminal ảo sẽ xuất hiện, một máy đại diện cho hệ thống splunk enterprise, một máy đại diện cho máy client thuộc splunk universal forwarder (UF), một máy đại diện cho máy attacker.

Để kiểm tra địa chỉ IP của 3 máy, gõ lệnh sau trên từng máy:

ifconfig

1.1.4.1 Triển khai và cấu hình hệ thống

1.1.4.1.1 Ở server – Triển khai và khởi tạo Splunk Enterprise

Đã tải sẵn bộ cài đặt Splunk ở client và server nên sinh viên chỉ cần giải nén và thực hiện lệnh. Trên server, sinh viên thực hiện lệnh sau:

sudo tar -xzf splunk-8.2.6-a6fe1ee8894b-Linux-x86_64.tgz -C /opt

sudo chown -R ubuntu:ubuntu /opt/splunk

Tiếp đến, lệnh ghi cấu hình OPTIMISTIC_ABOUT_FILE_LOCKING = 1 được thêm vào file splunk-launch.conf.

echo "OPTIMISTIC_ABOUT_FILE_LOCKING = 1" | sudo tee -a /opt/splunk/etc/splunk-launch.conf

sudo /opt/splunk/bin/splunk start --accept-license

Mục đích của dòng cấu hình này là yêu cầu Splunk bỏ qua kiểm tra khóa file trong môi trường ảo hóa hoặc container, vốn thường gây lỗi khi chạy trên nền tảng như Labtainer.

Cài đặt và đặt mật khẩu ngay trong quá trình khởi tạo xong thì thực hiện lệnh splunk status được dùng để bảo đảm dịch vụ splunkd đã hoạt động ổn định.

sudo /opt/splunk/bin/splunk status

Sinh viên sau đó mở trình duyệt (Firefox) tới địa chỉ http://127.0.0.1:8000, đăng nhập bằng tài khoản admin và mật khẩu đã đặt.

Ngay sau khi đăng nhập, một bước quan trọng khác là cấu hình Splunk lắng nghe dữ liệu gửi đến từ Universal Forwarder trên client.

sudo /opt/splunk/bin/splunk enable listen 9997 -auth <tài khoản>:<mật khẩu>

sudo netstat -tulpn | grep 9997

1.1.4.1.2 Ở client – Cài đặt Splunk Universal Forwarder (UF)

Phía client đóng vai trò nguồn gửi log về server nên cần triển khai Splunk Universal Forwarder.

sudo tar -xzf splunkforwarder-8.2.6-a6fe1ee8894b-Linux-x86_64.tgz -C /opt

sudo chown -R ubuntu:ubuntu /opt/splunkforwarder

Rồi đặt mật khẩu quản trị riêng :

sudo /opt/splunkforwarder/bin/splunk start --accept-license

Lệnh splunk status sau đó được dùng để kiểm tra xem UF đã khởi động đầy đủ hay chưa:

sudo /opt/splunkforwarder/bin/splunk status

Từ đó, ta thiết lập kết nối giữa UF và Splunk Enterprise:

sudo /opt/splunkforwarder/bin/splunk add forward-server 176.46.0.5:9997 -auth <tài khoản>:<mật khẩu>

1.1.4.1.3 Thiết lập log trên client – Tạo index trên server

Để Splunk có thể phân tích được hoạt động quét cổng, trước tiên client phải tạo ra một nguồn log ổn định, chứa đầy đủ các gói TCP SYN mà attacker gửi đến. Sinh viên tạo file /var/log/portscan_traffic.log ghi lại mọi gói TCP SYN mà attacker gửi đến.

sudo tcpdump -i eth0 -nn -tttt 'tcp[13] & 2 != 0' >> /var/log/portscan_traffic.log &

Bước tiếp theo là hướng Splunk UF theo dõi file đó. Điều này được thực hiện thông qua file cấu hình inputs.conf. Trong file này, một khối monitor được thêm vào:

```
[monitor:///var/log/portscan_traffic.log]
sourcetype = portscan:tcpdump
index = portscan_lab
disabled = false
```

Sau khi chỉnh cấu hình, UF phải được khởi động lại để nhận thay đổi.

Vì client gửi log vào index portscan_lab, Splunk Enterprise phải tạo index này trước.

sudo /opt/splunk/bin/splunk add index portscan_lab -auth <tài khoản>:<mật khẩu>

1.1.4.2 Phát hiện và phản ứng sự cố Port Scanning

1.1.4.2.1 Trên server : Phát hiện sự cố Port scanning

Ở phía attacker, sinh viên thực hiện lần lượt các kỹ thuật quét cổng để mô phỏng hành vi dò tìm dịch vụ của kẻ tấn công.

sudo nmap -sS -p- 176.46.0.7

sudo nmap -A 176.46.0.7

Ngay sau khi lệnh được thực thi, Splunk bắt đầu nhận thêm log mới từ client (tcpdump/syslog) và các sự kiện này sẽ xuất hiện trong index=portscan_lab.

Nhiệm vụ của người quản trị là mở giao diện Splunk Web và sử dụng SPL để phân tích dữ liệu, tìm ra các dấu hiệu bất thường của hoạt động quét cổng.

Đăng nhập Splunk Web tại http://127.0.0.1:8000 . Chọn mục Search & Reporting để xem dữ liệu trong index. Truy vấn cơ bản:

index=portscan_lab sourcetype=portscan:tcpdump

Để hiểu cuộc tấn công và phân biệt rõ ràng ai đang quét và họ đang quét những port nào, ta cần tách được ba thông tin quan trọng từ log: nguồn tấn công (src_ip), đích (dst_ip) và port bị quét (dst_port) :

```
index=portscan_lab sourcetype=portscan:tcpdump
| rex "IP (?<src_ip>[0-9]+\.[0-9]+\.[0-9]+\.[0-9]+)\.[0-9]+ > (?<dst_ip>[0-9]+\.[0-9]+\.[0-9]+\.[0-9]+)\.(?<dst_port>[0-9]+):"
| stats count by src_ip, dst_ip, dst_port
| sort - count
```

Ngoài giao diện web, Splunk cho phép truy vấn bằng CLI

sudo /opt/splunk/bin/splunk search "index=portscan_lab sourcetype=portscan:tcpdump" -maxout 5 -auth admin:Admin@123 | tee -a evidence.txt

1.1.4.2.2 Trên client : Kiểm soát sự cố tạm thời

Sau khi đã xác nhận hành vi quét cổng từ attacker, người quản trị cần thực hiện hành động containment, tức là ngăn chặn hoạt động tấn công đang diễn ra. Công cụ thích hợp nhất ở mức độ host là iptables.

Trước khi tạo rule, ta đảm bảo iptables đã được cài đặt. Việc chặn attacker được thực hiện bằng cách thêm rule DROP trên chuỗi INPUT để loại bỏ toàn bộ gói đến từ IP 176.46.0.3 — địa chỉ đã được Splunk xác định là nguồn tấn công.

sudo iptables -A INPUT -s 176.46.0.3 -j DROP

Ngay lập tức, các gói từ attacker sẽ bị chặn ở mức kernel. Khi attacker thử quét lại bằng nmap, quá trình quét sẽ chậm hơn, rất nhiều port báo filtered, và thời gian scan kéo dài bất thường.

1.1.4.2.3 Trên client : Diệt bỏ nguyên nhân (Eradication)

Bước trước chỉ chặn tạm thời IP tấn công, nhưng firewall có thể vẫn đang mở quá rộng. Ở bước này, ta tiến hành thiết lập bộ quy tắc firewall chặt chẽ hơn để giảm nguy cơ bị quét trong tương lai.

sudo iptables -F

Sau reset toàn bộ rule cũ, ta xây dựng bộ rule tối thiểu Bộ rule được xây dựng dựa trên hai nguyên tắc:

- Nguyên tắc mặc định từ chối (Default Deny) → Chỉ cho phép những gì thực sự cần.
- Nguyên tắc tối thiểu đặc quyền (Least Privilege) → Mỗi dịch vụ chỉ mở đúng phạm vi hoạt động.

Các rule bao gồm:

- Cho phép loopback
- Cho phép SSH từ đúng máy server duy nhất (176.46.0.5)

sudo iptables -A INPUT -p tcp -s 176.46.0.5 --dport 22 -j ACCEPT

Đây là rule quan trọng nhất trong bộ cấu hình. Chỉ duy nhất Splunk Server (176.46.0.5) được phép SSH vào Client , để giới hạn tuyệt đối bề mặt tấn công từ bên ngoài.

- Cho phép lưu lượng HTTP vào Client trong bài lab

sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT

- Cho phép các gói thuộc về một phiên kết nối hợp lệ mà máy Client đã khởi tạo OUTBOUND.
- Cuối cùng , mọi traffic không khớp với bất kỳ rule nào phía trên đều bị loại bỏ.

sudo iptables -A INPUT -j DROP

➔ Đảm bảo hệ thống chỉ chạy đúng dịch vụ cho phép.

Bộ rule này hạn chế tối đa bề mặt tấn công: chỉ những dịch vụ cần thiết mới mở, còn lại bị loại bỏ.

1.1.4.2.4 Trên client : Thu thập chứng cứ

Sau khi tấn công đã được kiểm soát và firewall đã được siết chặt, bước tiếp theo là thu thập chứng cứ để phục vụ phân tích hoặc báo cáo. Trên client, log portscan đã được lưu trong /var/log/portscan_traffic.log, và ta cần nén lại rồi tạo mã băm để đảm bảo tính toàn vẹn.

mkdir -p ~/ir-backup

sudo tar czf ~/ir-backup/portscan_logs.tar.gz /var/log/portscan_traffic.log

sha256sum ~/ir-backup/portscan_logs.tar.gz | tee ~/ir-backup/portscan_logs.sha256

1.1.4.3 Theo dõi hậu sự cố

Sau khi sự cố đã được kiểm soát và log đã được lưu lại, bước tiếp theo trong quy trình phản ứng sự cố là thiết lập cơ chế giám sát tự động để đảm bảo rằng nếu tấn công tương tự xảy ra trở lại, hệ thống sẽ chủ động cảnh báo. Việc thiết lập cảnh báo (alert) trong Splunk giúp quá trình phát hiện được tự động hóa hoàn toàn, không phụ thuộc vào việc người vận hành phải liên tục quan sát giao diện.

```
index=portscan_lab sourcetype=portscan:tcpdump
| rex "IP (?<src_ip>[0-9]+\.[0-9]+\.[0-9]+\.[0-9]+)\.[0-9]+ > (?<dst_ip>[0-9]+\.[0-9]+\.[0-9]+\.[0-9]+)\.(?<dst_port>[0-9]+):"
| bin _time span=1m
| stats dc(dst_port) AS uniq_ports count AS total_events BY src_ip,_time
| where uniq_ports>=50
| sort - _time
```

Câu truy vấn được dùng trong alert phải thỏa mãn 3 yêu cầu:

- Đếm số port khác nhau mà một src_ip đã quét.
- Phân tích theo từng phút (vì port scan có tính chất bùng phát theo thời gian).
- Lọc ra những IP có uniq_ports ≥ 50, được xem là hành vi scanning rõ ràng.

Sau khi chạy truy vấn và đảm bảo nó trả kết quả đúng, ta lưu nó thành alert bằng cách chọn Save As → Alert.

- Phần thông tin Alert

- Title: Port scan detected (portscan_lab)
- Description: mô tả mục tiêu <tùy chọn>
- Quyền truy cập có thể để mặc định (Private).
- Alert Type chọn Scheduled
- Lịch chạy: Run on Cron Schedule, điền ô nhập : * * * * *

- Phần Trigger (điều kiện bắn cảnh báo)
  - Trigger alert when: Number of Results → is greater than: 0
  - Chế độ Trigger: Once – mỗi lần Splunk chạy truy vấn mà thấy kết quả, sẽ bắn một alert.
- Phần Actions (thao tác khi alert kích hoạt)
  - Event: Port scan detected: src_ip=$result.src_ip$, uniq_ports=$result.uniq_ports$, total_events=$result.total_events$
  - Sourcetype: portscan_alert
  - Index: portscan_lab

Để kiểm tra alert vận hành đúng, attacker tạo lại tấn công bằng lệnh nmap.

Vì alert được cấu hình chạy mỗi phút, ta đợi khoảng 1–2 phút để Splunk xử lý dữ liệu và đánh giá điều kiện kích hoạt. Khi alert hoạt động, có hai cách kiểm tra:

Trên Splunk Web, ta vào Search rồi truy vấn kiểm tra xem đã có bản ghi chưa:

index=portscan_lab sourcetype=portscan_alert

Trên màn hình lệnh CLI :

sudo /opt/splunk/bin/splunk search 'index=portscan_lab sourcetype=portscan_alert' -auth admin:Admin@123 -maxout 20 | tee -a evidence.txt

1.1.4.4 Kết thúc bài lab:

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

stoplab

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.

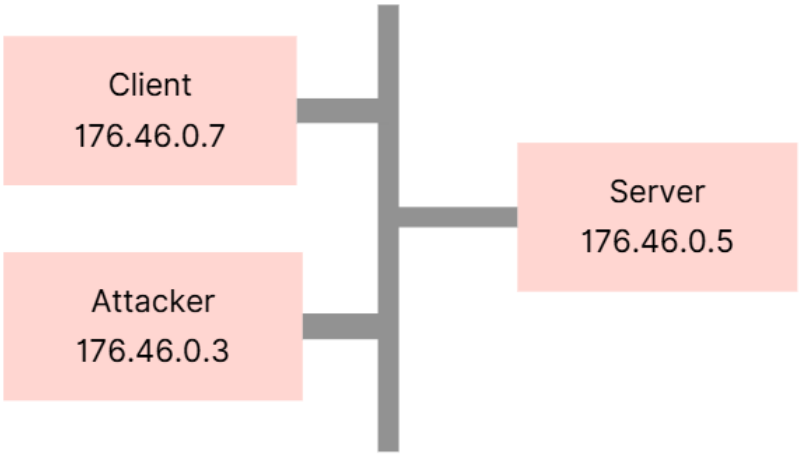
Trong quá trình làm bài sinh viên cần thực hiện lại bài lab, dùng câu lệnh:

labtainer -r idr_splunk_portscanning

1.1.5 Thiết kế bài thực hành

1.1.5.1 Cấu hình docker

- Trong môi trường máy ảo Ubuntu, sử dụng docker tạo ra 3 container: : 1 container “client”, 1 container “server, 1 container “attacker”.
- Tạo mạng có cấu hình:
    - Subnet: 176.46.0.0/24
    - External Gateway: 176.46.0.1
- Cấu hình docker gồm có:
    - Server: Cấu hình cho máy server
        - Tên máy: server
        - Địa chỉ trong mạng LAN: 176.46.0.5
        - Gateway: 176.46.0.1
    - Client: Cấu hình cho máy client
        - Tên máy: client
        - Địa chỉ trong mạng LAN: 176.46.0.7
        - Gateway: 176.46.0.1
    - Attacker: Cấu hình cho máy tấn công 2 sự cố
        - Tên máy: attacker
        - Địa chỉ trong mạng LAN: 176.46.0.3
        - Gateway: 176.46.0.1



Hình 1 Topo mạng bài idr_splunk_portscanning

- config: lưu cấu hình hoạt động của hệ thống

- dockerfiles: mô tả cấu hình của các container, gồm:
    - server: sử dụng các thư viện mặc định hệ thống, cập nhật hệ thống và cài sẵn Splunk Enterprise 8.2.6.
    - client: sử dụng các thư viện mặc định hệ thống, cập nhật hệ thống và cài sẵn Splunk Universal Forwarder.
    - attacker: sử dụng các thư viện mặc định hệ thống và cập nhật hệ thống, cài đặt thêm công cụ nmap.

1.1.5.2 Các nhiệm vụ cần thực hiện

- Vận hành Splunk server để thu thập và phân tích log tập trung.
- Vận hành Splunk Forwarder trên client để gửi log về server.
- Ghi nhận địa chỉ IP máy tấn công trong file bằng chứng trên server.
- Cấu hình tường lửa trên client để xử lý lưu lượng từ IP 176.46.0.3.
- Cấu hình tường lửa trên client để xử lý lưu lượng liên quan IP 176.46.0.5.
- Sao lưu log/bằng chứng và kiểm tra tính toàn vẹn bằng mã băm.
- Ghi rõ trường `src_ip` của lưu lượng tấn công trong file bằng chứng phục vụ phân tích.

1.1.5.3 Các kết quả cần đạt được

Mỗi nhiệm vụ nhỏ sẽ được chia ra thành các mục chấm điểm để xác nhận sinh viên đã làm đúng các bước và hoàn thành bài thực hành hay chưa. Vì vậy, hệ thống sẽ ghi nhận các thao tác, sự kiện được mô tả theo bảng dưới đây để chấm điểm cho sinh viên:

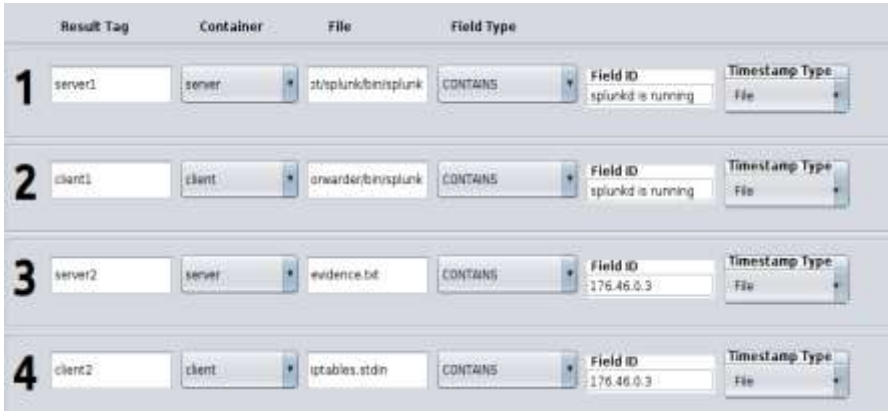
Bảng 1. Bảng result của bài idr_splunk_portscanning

Result Tag	Container	File	File Type	Field ID	Timestamp Type
server1	server	/opt/splunk/bin/splunk	CONTAINS	splunkd is running	File
client1	client	/opt/splunkforwarder/bin/splunk	CONTAINS	splunkd is running	File
server2	server	evidence.txt	CONTAINS	176.46.0.3	File
client2	client	iptables.stdin	CONTAINS	176.46.0.3	File
client3	client	iptables.stdin	CONTAINS	176.46.0.5	File

client4	client	sha256sum.stdin	CONTAINS	backup	File
server3	server	evidence.txt	CONTAINS	src_ip=176.46.0.3	File

1.1.6 Cài đặt và cấu hình các máy ảo

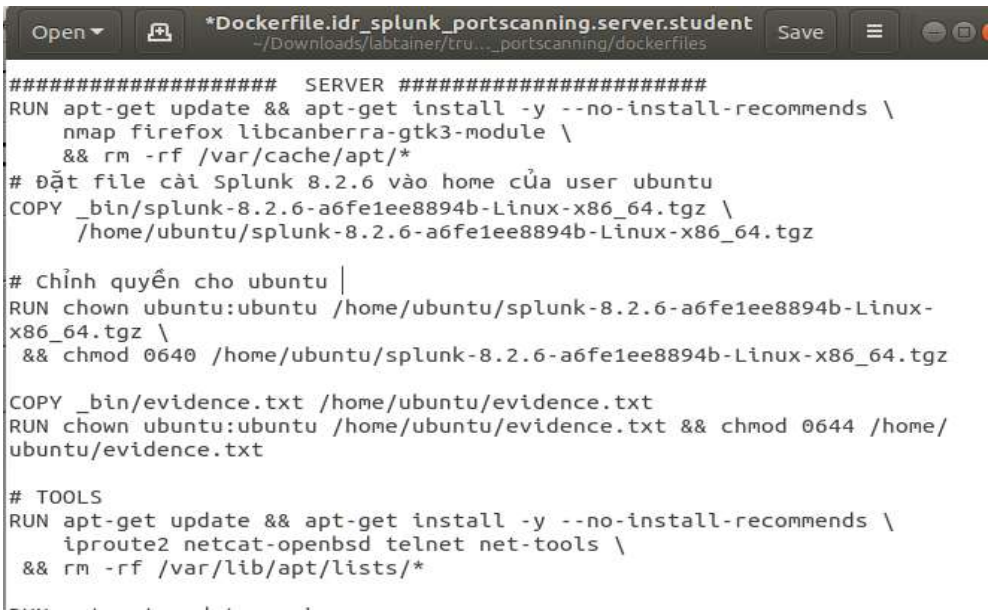
Hình 2 Giao diện lab idr_splunk_portscanning



Hình 3 Bảng Result bài lab idr_splunk_portscanning

Hình 4 Dockerfiles của máy attacker

Hình 5 Dockerfiles của máy client



Hình 6 Dockerfiles của máy server

1.1.7 Tích hợp và triển khai

1.1.7.1 Docker Hub

Hình 7 Bài thực hành được lưu trữ trên docker hub

1.1.7.2 Github

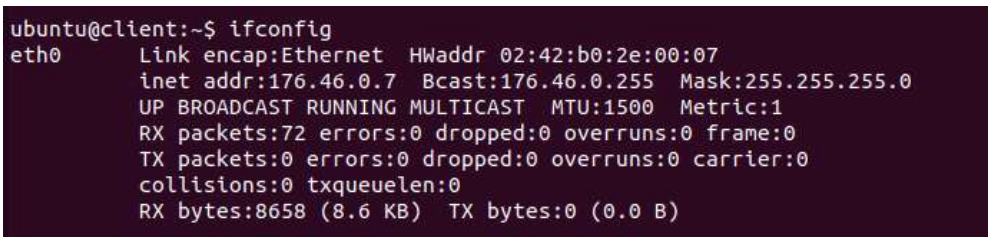
Link github:
https://github.com/anhdnmit/do_an_tot_nghiep/tree/main/idr_splunk_webscan

Hình 8 Đẩy file imodule.tar lên github

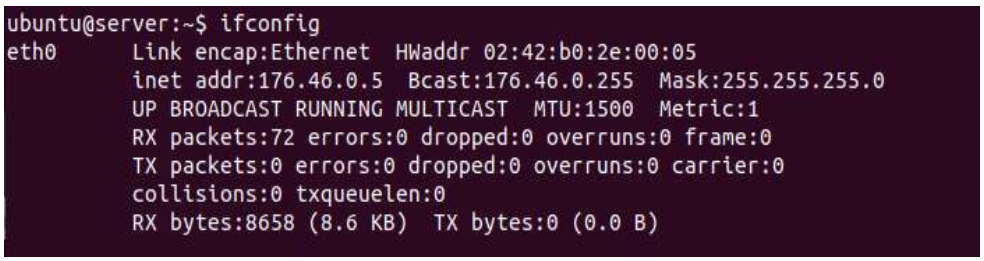
1.1.8 Thử nghiệm và đánh giá

Bài thực hành đã được xây dựng thành công. Bây giờ ta sẽ đánh giá về 1 trong các nhiệm vụ bài lab yêu cầu.

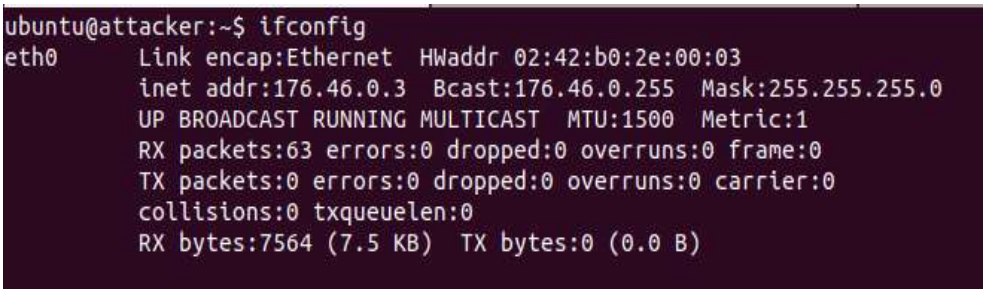
Trước hết, ta kiểm tra ip trên 3 máy.



Hình 9 IP client



Hình 10 IP server



Hình 11 IP attacker

Ngoài 2 nhiệm vụ liên quan về mặt "Triển khai cấu hình" ở trên server và client cùng với nhiệm vụ "Ghi nhận địa chỉ IP nguồn tấn công trong file bằng chứng trên server" trong giai đoạn "Phát hiện sự cố" đã làm , giờ ta đánh giá nhiệm vụ 5 "Cấu hình tường lửa trên client để xử lý lưu lượng liên quan IP 176.46.0.5." liên quan đến giai đoạn Eradication (diệt bỏ nguyên nhân)

Ở bước này, mục tiêu là siết chặt firewall trên client sao cho chỉ cho phép những kết nối cần thiết (đặc biệt là SSH từ IP 176.46.0.5 – máy server), và chặn những lưu lượng không mong muốn còn lại. Việc này giúp giảm bề mặt tấn công và cắt đi con đường kẻ tấn công lợi dụng.

Đầu tiên là xây dựng lại chính sách tường lửa. Ta cho phép traffic trên loopback (lo), các kết nối nội bộ trong chính máy client, để tránh làm hỏng các dịch vụ chạy cục bộ.

Tiếp theo, sinh viên xử lý lưu lượng liên quan tới IP 176.46.0.5.

Ở đây, IP này được xem như máy server hợp lệ, nên ta chỉ cho phép SSH từ địa chỉ này truy cập vào client. Điều này giúp đảm bảo rằng việc quản lý từ xa vẫn thực hiện được, nhưng các IP khác không thể tùy tiện SSH vào máy:

sudo iptables -A INPUT -p tcp -s 176.46.0.5 --dport 22 -j ACCEPT

Song song với việc bảo vệ SSH, ta vẫn cần giữ cho dịch vụ web hoạt động bình thường với người dùng. Vì vậy, ta mở cổng 80 để cho phép các kết nối HTTP đi vào:

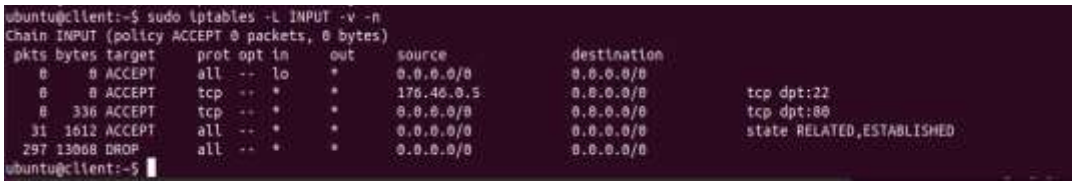
sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT

Để không làm gián đoạn các kết nối đã được thiết lập trước đó, ta thêm một rule cho phép các gói thuộc những kết nối đang tồn tại hoặc có liên quan (ESTABLISHED, RELATED). Quy tắc này đảm bảo hệ thống vẫn ổn định khi có các phiên giao tiếp đang diễn ra

Sau khi đã khai báo đầy đủ các trường hợp "được phép", ta đặt một rule chặn mặc định cho mọi lưu lượng khác đi vào chain INPUT. Rule này đóng vai trò "chốt chặn cuối", đảm bảo toàn bộ traffic không khớp với các điều kiện ở trên, bao gồm các nỗ lực kết nối trái phép, đều bị drop:

sudo iptables -A INPUT -j DROP

Cuối cùng, để kiểm tra lại cấu hình vừa áp dụng và quan sát thứ tự cũng như thống kê lưu lượng qua từng rule, ta liệt kê chi tiết chain INPUT:



Hình 12 Cấu hình áp dụng cho bài lab idr_splunk_portscanning

Hình 13 Checkwork bài lab idr_splunk_portscanning