Đại học Quốc gia thành phố Hồ Chí Minh
Trường Đại học Công nghệ Thông tin
Khoa mạng máy tính và truyền thông
Bộ môn An toàn thông tin

1

Lab

PHỤC VỤ MỤC ĐÍCH GIÁO DỤC
FOR EDUCATIONAL PURPOSE ONLY

Packet Sniffing & Spoofing

Thực hành môn An toàn mạng

Tháng 9/2024

Lưu hành nội bộ

<Nghiêm cấm đăng tải trên internet dưới mọi hình thức>

A. TỔNG QUAN

1. Mục tiêu

- Hiểu được khái niệm về Packet Sniffing và Packet Spoofing.
- Tìm hiểu cách thức hoạt động của các công cụ liên quan đến Packet Sniffing & Spoofing.
- Thực hành phân tích và thu thập dữ liệu mạng sử dụng Packet Sniffing tool.
- Thực hành việc giả mạo (spoof) các gói tin bằng Scapy.

2. Thời gian thực hành

- Thực hành tại lớp: 5 tiết tại phòng thực hành.
- Hoàn thành báo cáo kết quả thực hành: tối đa 13 ngày.

B. CHUẨN BỊ MÔI TRƯỜNG

- Cài đặt máy ảo Seed Ubuntu 20.04 (https://seedsecuritylabs.org/labsetup.html) trên Virtual Box
- Tải xuống các tập tin được đính kèm theo bài lab và thực hiện phần chuẩn bị môi trường tại phần C.2.a

C. THỰC HÀNH

1. Packet Sniffing

Packet Sniffing là quá trình theo dõi và thu thập các gói tin truyền qua một mạng máy tính. Những gói tin này mang thông tin về việc trao đổi dữ liệu giữa các thiết bị, có thể chứa các nội dung như địa chỉ IP, thông tin đăng nhập, email, hoặc các dữ liệu nhạy cảm khác. Các công cụ được sử dụng trong quá trình này được gọi là network sniffer, packet sniffer, hay network analyzer. Quản trị viên mạng thường sử dụng các công cụ này để giám sát lưu lượng mạng, khắc phục sự cố hoặc phát hiện lỗ hổng bảo mật. Tuy nhiên, các kẻ tấn công cũng có thể lợi dụng những công cụ này để thực hiện các cuộc Tấn công nghe lén (eavesdropping) và đánh cắp thông tin nhạy cảm như mật khẩu, số thẻ tín dụng hoặc thông tin cá nhân khác hoặc Tấn công man-in-the-middle chặn và thay đổi thông tin truyền qua mạng nhằm phục vụ các các mục đích xấu.

Ở phần này, chúng ta sẽ sử dụng các công cụ packet sniffing để thu thập, phân tích lưu lượng mạng để phục vụ quá trình điều tra tấn công.

Sử dụng các file pcap được cung cấp trong bài lab và Wireshark, thực hiện các task dưới đây:

Task 1: File exfil-1.pcap ghi lại lưu lượng mạng của một tổ chức. Trong đó, người quản trị viên đang muốn tìm kẻ tấn công đang cố gắng gửi dữ liệu ra ngoài. Biết rằng kẻ tấn công

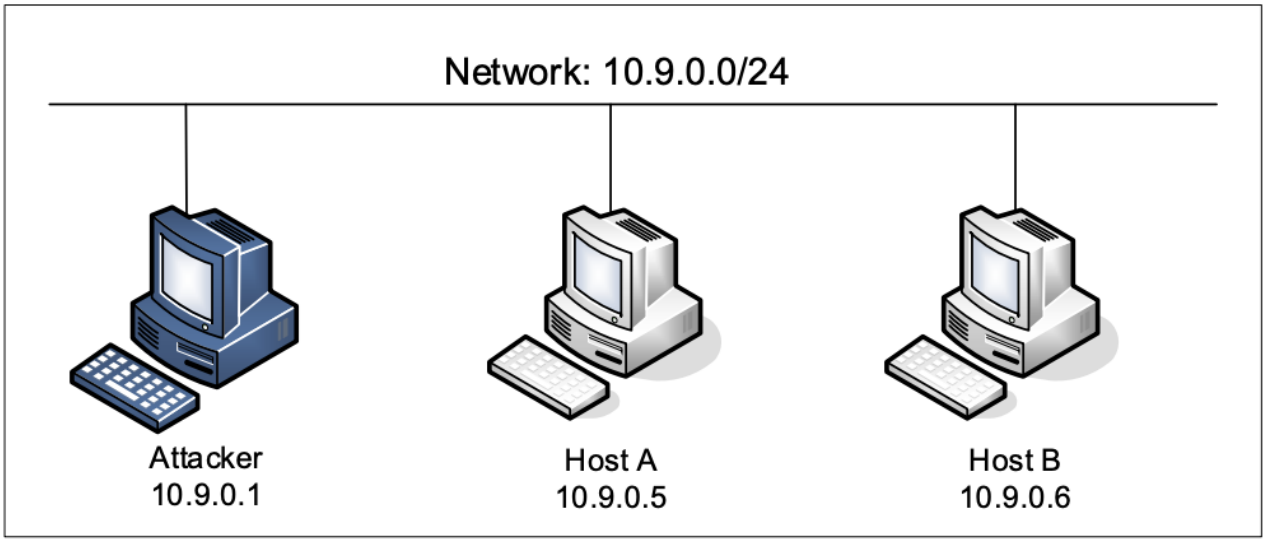
không sử dụng giao thức TCP và UDP. Hãy giúp quản trị viên này tìm ra thông điệp mà kẻ tấn công gửi đi.

Task 2: Trong task này, chúng ta sẽ cố gắng tìm ra những tin tặc đã truy cập vào camera an ninh trong toà nhà thông qua lỗ hổng bảo mật trên máy chủ web. Hãy dựa vào thông tin về lỗ hổng bảo mật tại link https://www.coresecurity.com/core-labs/advisories/d-link-ip-cameras-multiple-vulnerabilities, phân tích file attack.pcap, tìm ra IP của kẻ tấn công và mô tả lại quá trình tấn công.

2. Packet Spoofing

Ở phần này, chúng ta sẽ thực hiện sử dụng Scapy để đánh hơi và giả mạo các gói tin. Để làm được điều này, chúng ta cần tạo môi trường thử nghiệm sử dụng 3 container với mô hình mạng được mô tả ở hình 1.

a. Chuẩn bị môi trường



Hình 1 Mô hình mạng

Cài đặt máy ảo Seed Ubuntu dựa trên hướng dẫn sau: https://github.com/seed-labs/seed-labs/blob/master/manuals/vm/seedvm-manual.md

Sau khi cài đặt thành công, truy cập vào máy ảo Seed Ubuntu với mật khẩu "dees". Trên máy ảo Seed Ubuntu, tải và giải nén file Labstepup.zip, sau đó truy cập vào thư mục Labsetup. Sử dụng các câu lệnh docker sau để xây dựng mô hình mạng ở hình 1.

```
$ docker-compose build       # Build the container image
$ docker-compose up          # Start the container
```

Lúc này các container đã được khởi chạy, chúng ta cần thực hiện truy cập vào shell của các container để thực thi lệnh. Trước tiên, chúng ta cần sử dụng lệnh "docker ps" để tìm ID của một container, sau đó sử dụng "docker exec" để khởi động shell trên container đó.
Mở một terminal mới và thực thi các lệnh sau:

```
$ docker ps --format "{{.ID}} {{.Names}}"

951e36a9531f seed-attacker

c9c520323b7f hostB-10.9.0.6

1505f05277d9 hostA-10.9.0.5

$ docker exec -it 951e36a9531f /bin/bash        # truy cập vào shell trên container seed attacker
root@docker-desktop:/#
```

Trong phần thực hành này, chúng ta sẽ sử dụng attacker container làm máy tấn công. Để thực hiện được các yêu cầu của phần này, chúng ta sẽ cần thực thi các đoạn code bên trong máy attacker. Với môi trường container, có nhiều bất tiện khi thực hiện soạn thảo code, do đó, một thư mục ./volumes đã được mount sẵn với máy host Seed Ubuntu. Chúng ta có thể viết code cho các task trong phần 2 này vào thư mục ./volumes này và sử dụng các trình soạn thảo trên Seed Ubuntu để chỉnh sửa code.
Kẻ tấn công cần có khả năng đánh hơi, bắt các gói tin, nhưng việc bắt gói tin bên trong container sẽ gặp nhiều vấn đề vì container được gắn với các switch ảo, do đó, nó chỉ có thể nhìn thấy lưu lượng truy cập của chính nó và không thể xem được các gói tin từ các container khác. Để giải quyết vấn đề này, attacker được cài đặt ở chế độ network_mode: host , cho phép xem tất cả các lưu lượng truy cập.
Tiếp theo đây, chúng ta sẽ sử dụng Scapy để thực hiện đánh hơi và giả mạo gói tin.

b. Sử dụng Scapy để đánh hơi và giả mạo gói tin

Scapy là một thư viện Python mạnh mẽ dùng để phân tích, tạo, gửi và giả mạo gói tin trên mạng. Scapy cho phép người dùng có thể lịch hoạt trong việc tạo ra các gói tin tuỳ chính và kiểm tra cách hệ thống phản ứng với chúng. Chúng ta cũng có thể tích hợp các chức năng của Scapy vào các chương trình riêng của mình.

```
#!/usr/bin/env python3

from scapy.all import *

def print_pkt(pkt):

          pkt.show()

pkt = sniff(iface='br-c93733e9f913', filter='icmp', prn=print_pkt)
```

Task 3: Đoạn code trên là ví dụ cho việc sử dụng scapy đánh hơi các gói tin trên interface br-c93733e9f913. Hàm print_pkt(pkt) thực hiện in các thông tin của gói tin đánh hơi được. Thực thi đoạn code trên và mô tả kết quả quan sát được.
Lưu ý: sử dụng lệnh ifconfig tại container attacker để tìm ra interface gắn với mạng 10.9.0.0/24 và thay thế vào trường *iface='br-c93733e9f913'* trong đoạn code trên.

```
// Make the program executable

root@docker-desktop:/# chmod a+x sniffer.py

// Run the program with the root privilege

root@docker-desktop:/# sniffer.py
```

Sau khi chạy lệnh sniffer.py, có thể bạn sẽ không thấy thông tin nào được in ra màn hình, bởi vì đang không có gói tin nào được truyền đi trong mạng 10.9.0.0/24. Do đó, để chương trình của chúng ta có thể bắt được gói tin trong mạng và in ra màn hình, cần tạo ra các lưu lượng mạng. Hãy thử truy cập vào một container và tạo một lệnh ping tới một máy khác trong mạng và quan sát lại kết quả.

Task 4: Thông thường, khi đánh hơi các gói tin, chúng ta chỉ quan tâm đến một số loại gói tin nhất định. Hãy sử dụng các bộ lọc của Scapy để thu thập các gói tin theo từng yêu cầu sau:

- Chỉ bắt những gói tin ICMP
- Bắt các gói tin đến từ một IP cụ thể có cổng đích là 23

Là một công cụ giả mạo gói tin, Scapy cho phép chúng ta đặt các trường của gói IP thành các giá trị tuỳ ý. Ở task này, chúng ta sẽ giả mạo các gói ICMP echo request và gửi chúng đến một máy ảo khác trên cùng một mạng.

Task 5: Chỉnh sửa lại đoạn code dưới đây để tạo một gói tin ICMP echo request giả mạo. Dùng Wireshark để quan sát xem gói tin yêu cầu có được chấp nhận hay không. Nếu được chấp nhận, gói tin phản hồi sẽ được gửi đến địa chỉ IP giả mạo.

```
#!/usr/bin/env python3

from scapy.all import *

a = IP ()

a.src = '<IP gia mao>'

a.dst = '<IP dich>'

b= ICMP ()
```

```
p = a/b

send (p)
```

Bây giờ, chúng ta sẽ kết hợp kỹ thuật bắt gói tin và giả mạo để triển khai thành một chương trình hoàn chỉnh

Task 6: Viết chương trình sniff_spoof.py thực hiện sniff và spoof gói tin ICMP. Cụ thể:
Sử dụng 2 container đã được tạo, trong đó có container attacker. Khi thực hiện một lệnh ping tới IP X, sẽ có một gói tin ICMP echo request được tạo ra. Nếu X là địa chỉ hợp lệ, chương trình ping sẽ nhận được gói echo reply và in ra phản hồi.
Đoạn code của bạn cần được chạy trên attacker container. Bất cứ khi nào nó nhận thấy ICMP echo request, bất kể địa chỉ IP đích là gì, chương trình của bạn sẽ gửi phản hồi với một gói tin giả mạo.
Do đó, bất kể X có là địa chỉ hợp lệ hay không, lệnh ping sẽ luôn nhận được phản hồi cho biết X hợp lệ.
Kiểm tra chương trình với một địa chỉ IP hợp lệ và một địa chỉ IP không tồn tại để kiểm tra tính đúng của chương trình.

D. YÊU CẦU & ĐÁNH GIÁ

- Sinh viên tìm hiểu và thực hành theo hướng dẫn, thực hiện theo nhóm đã đăng ký.
- Nộp báo cáo kết quả gồm chi tiết những việc (Report) mà nhóm đã thực hiện, quan sát thấy và kèm ảnh chụp màn hình kết quả (nếu có); giải thích cho quan sát (nếu có).
- Báo cáo:
  - File .PDF. Tập trung vào nội dung, không mô tả lý thuyết.
  - Đặt tên theo định dạng: [Mã lớp]-LabX_MSSV1_MSSV2.
  - Ví dụ: [NT140.P12.ANTT.1]-Lab1_2252xxxx_2252yyyy.
  - Nếu báo cáo có nhiều file, nén tất cả file vào file .ZIP với cùng tên file báo cáo.
  - Nộp file báo cáo trên theo thời gian đã thống nhất tại courses.uit.edu.vn.

Bài sao chép, trễ, ... sẽ được xử lý tùy mức độ vi phạm.

HẾT

Chúc các bạn hoàn thành tốt!