**Nội dung và hướng dẫn bài thực hành**

***Mục đích***

Giúp sinh viên tiếp cận với một số cách thăm dò mạng và phát hiện ra dấu hiệu của chúng bằng wireshark.

**Yêu cầu đối với sinh viên**

Có kiến thức cơ bản về Wireshark

**Nội dung thực hành**

Khởi động bài lab:

Vào terminal, gõ:

*labtainer -r net-recon-sign*

*(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 terminal student ảo sẽ xuất hiện.

Gõ lệnh: *wireshark*để khởi động wireshark.

Sau đó sinh viên tìm hiểu về các loại thăm dò sau đó trả lời câu hỏi:

* Task 1: TCP-Half\_Open\_Scan:
* Là một kỹ thuật thăm dò bằng cách quét cổng (port scanning) nhằm xác định cổng nào đang mở trên một máy mục tiêu mà không hoàn thành kết nối TCP (bắt tay 3 bước).
* Bộ lọc:

*tcp.flags.syn == 1*

(Bộ lọc sẽ trả về những gói tin chứa yêu cầu SYN và phản hồi SYN)

* Cổng mở, máy chủ sẽ trả về gói tin (SYN, ACK)
* Cổng đóng: máy chủ trả về gói tin (RST, ACK) có màu đỏ

* Để kiểm tra những cổng mở cần dùng bộ lọc xem được gói tin (SYN, ACK):

*tcp.flags.syn == 1 && tcp.flags.ack == 1*

* Nếu phát hiện lượng lớn gói tin (SYN, ACK) của 1 địa chỉ MAC trong cùng khoảng thời gian thì có thể coi đây là 1 cuộc thăm dò cổng TCP.

* **Câu hỏi**: Cổng lớn nhất của đị chỉ IP 10.10.10.12 có phản hồi là bao nhiêu?

* **Task 2:** **ARP-Sweep:**
* Kẻ tấn công gửi các gói ARP-request yêu cầu địa chỉ MAC của các máy trong cùng một mạng. Hành động thăm dò này thường để chuẩn bị cho các cuộc tấn công như MiTM.
* Để kiểm tra các gói tin arp yêu cầu, sử dụng bộ lọc:

*arp.opcode == 1*

* Vì không biết địa chỉ MAC của các máy trong mạng nên kẻ tấn công sẽ gửi đến địa chỉ MAC 00:00:00:00:00:00 (Gửi đến toàn bộ mạng). Kết hợp 2 bộ lọc này ta có thể tìm ra những máy khả nghi:

*arp.dst.hw\_mac==00:00:00:00:00:00*

* **Câu hỏi:** Địa chỉ MAC 00:15:5d:14:3c:7f gửi nhiều hơn 00:15:5d:14:3c:7c bao nhiêu gói tin?

* Task 3: OS-Fingerprinting:
* Là kỹ thuật thăm dò thông tin về hệ điều hành của nạn nhân, từ đó kẻ tấn công có thể nhắm vào những lỗ hổng trong hệ điều hành đó.
* OS-Fingerprinting thu thập dữ liệu bằng cách gửi gói tin TCP SYN và xem phản hồi, mỗi hệ điều hành sẽ có những đặc điểm riêng nằm trong tcp flag hoặc các thông số trong header.
* OS-Fingerprinting thường sử dụng giá trị nhỏ, vì chúng không cần dữ liệu thực tế mà chỉ muốn kiểm tra dữ liệu từ gói tin phản hồi.
* Sử dụng bộ lọc:

*tcp.flags==0x02 && tcp.window\_size<1025*

* Sự tồn tại của gói tin này được cho là 1 cuộc thăm dò lấy thông tin của hệ điều hành.
* Câu hỏi: OS-Fingerprinting: Có bao nhiêu gói tin được gửi đến cổng 21

* **Task 4:** **Bruteforce:**
* Là cuộc tấn công trên máy chủ FTP, kẻ tấn công thử tài khoản và mật khẩu nhiều lần để chiếm được quyền truy cập vào máy chủ. Nếu thử thất bại, máy chủ sẽ gửi gói tin chứa mã phản hồi 220.
* Bộ lọc:

*ftp.response.code==220*

* Một lượng lớn lần đăng nhập thất bại có thể coi là tấn công bruteforce.
* **Câu hỏi:** Có bao nhiêu gói tin đăng nhập thất bại trong 10s đầu

* **Task 5:** **Pony Malware:**
* Là phần mềm đánh cắp dữ liệu từ máy bị nhiễm.
* Sử dụng bộ lọc:

*http.request.method == "POST”*

* Tìm những đường link đáng ngờ (có tên miền lạ)
* Sử dụng VirusTotal để kiểm tra đường link mà máy đã kết nối

* **Câu hỏi:** Máy chủ có IP 10.2.1.101 kết nối với bao nhiêu đường link(khác nhau) nhiễm mã độc?

* Kiểm tra kết quả:
* Sinh viên tạo 1 file và lưu kết quả vào 5 dòng khác nhau với định dạng:

Task *x:*(*Kết quả*)

* Lệnh tạo file:

*nano tên\_file*

* Thực hiện biên dịch chương trình result.c và chạy:

*gcc result.c -o result*

*./result*

* Sau khi chạy chương trình, nhập tên file chứa đáp án vào để kiểm tra kết quả.

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.