Lab
6

BÁO CÁO BÀI THỰC HÀNH SỐ 6
Linux Firewall Exploration

Môn học: An toàn mạng máy tính
Lớp: NT101.O11.MMCL.1

Giảng viên hướng dẫn	Nguyễn Xuân Hà
Sinh viên thực hiện	Lê Huỳnh Quang Vũ (21522797)
	Nguyễn Trọng Phúc (21522476)
	Trần Nguyễn Quốc Bảo (21521865)
Mức độ hoàn thành	Hoàn thành
Thời gian thực hiện	06/12/2023 – 12/12/2023
Tự chấm điểm	9/10

A. CÁC BƯỚC THỰC HÀNH

B. TRẢ LỜI CÁC CÂU HỎI

Từ máy A, sử dụng lệnh sau để thiết lập SSH tunnel:

```
$ ssh -fN -L 8000:localhost:23 VM_B_username@VM_B_IP

$ telnet localhost 8000
```

➲ *1. Trình bày ý nghĩa các tham số sử dụng trong 2 lệnh thiết lập tunnel và kết nối telnet ở trên.*

```
$ ssh -fN -L 8000:localhost:23 ubuntu@192.168.103.10
```

Trong đó:

- ssh: là lệnh để kết nối đến một máy chủ từ xa qua giao thức SSH
- -f: không thực hiện bất kỳ lệnh cụ thể sau khi kết nối.
- -L 8000:localhost:23: Tạo một kênh chuyển tiếp cục bộ (local forward) từ cổng 8000 của máy cục bộ đến cổng 23 của localhost trên máy chủ từ xa.
- ubuntu@192.168.103.10: Tên người dùng và địa chỉ IP của máy chủ từ xa.

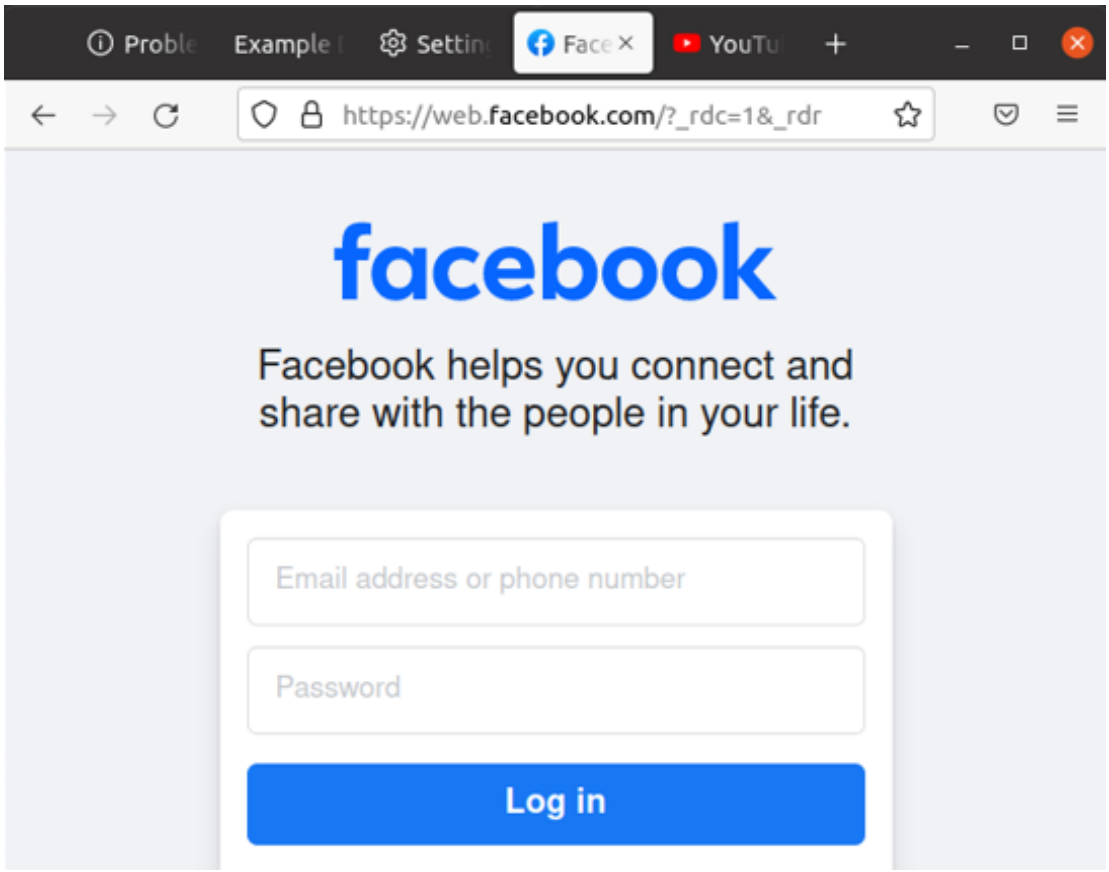
```
$ telnet localhost 8000
```

- Lệnh này để kết nối telnet đến máy B thông qua tunnel đã thiết lập.
- Từ đó ta thấy, muốn vượt qua sự kiểm soát của firewall đã thiết lập không cho máy VM A kết nối đến VM B thì ta đã thiết lập tunnel để máy VM A sử dụng giao thức ssh đến một kênh chuyển tiếp cục bộ thông qua port 8000, từ đó kênh chuyển tiếp này sẽ sủ dụng giao thức telnet qua port 23 để tiến hành telnet sang máy VM B.

➲ *2. Khi sử dụng lệnh telnet, thực chất các gói tin này có đi qua máy Firewall không? Nếu có, nguyên nhân tại sao Firewall không việc sử dụng telnet này? Nếu không, thì kết nối từ máy A đến máy B như thế nào để không đi qua máy Firewall?*

- Gói tin thực chất đi qua Firewall.
- Nguyên nhân Firewall ko chặn vì chúng ta đã mở port 23 để telnet và lấy port 8000 để alias gói tin về port 23.
- Do đó khi máy nội bộ tiến hành telnet đến máy WAN, thì sẽ telnet đến port 8000 của máy WAN như câu lệnh trên và từ port 8000 đó ánh xạ về port 23 trong nội bộ máy WAN do đó không phải chịu ảnh hưởng từ Firewall.

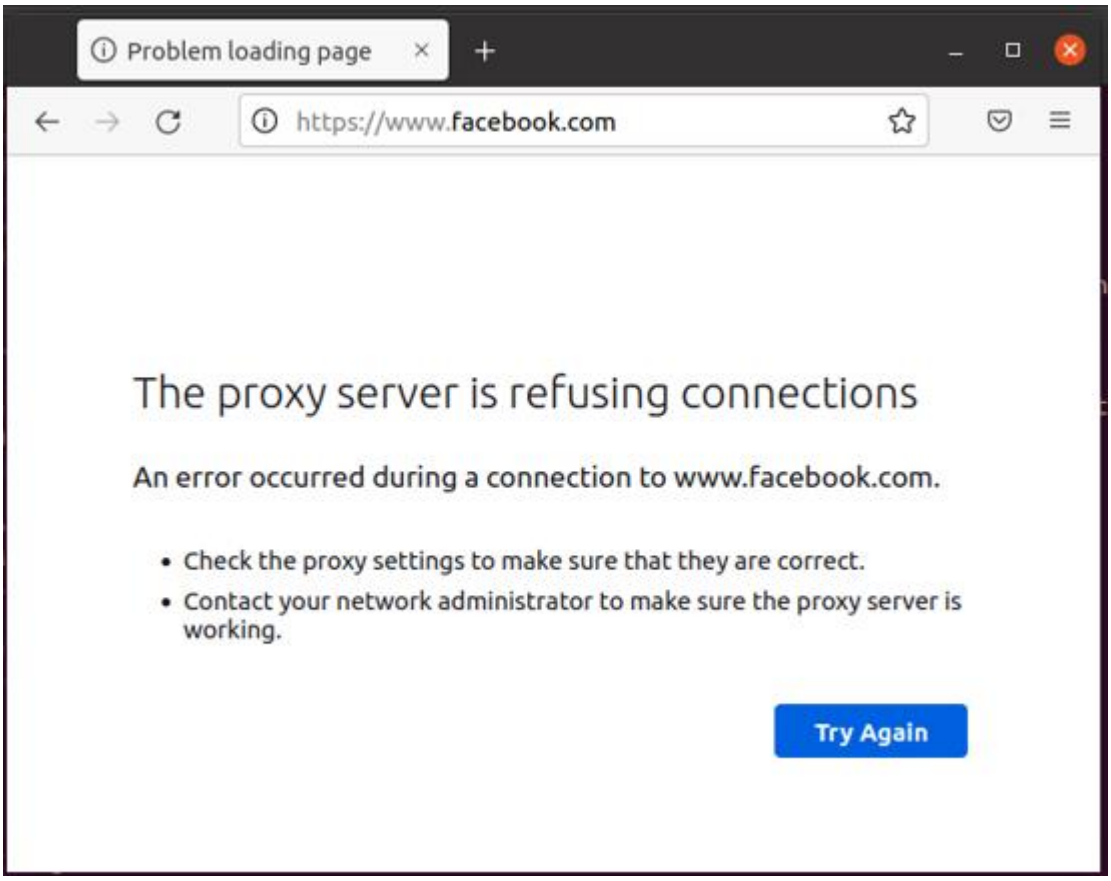
➲ *3. Truy cập website www.facebook.com. Mô tả quá trình bạn quan sát được.*



Truy cập facebook.com

Mô tả: trang web có thể truy cập bình thường.

➲ *4. Thực hiện ngắt SSH Tunnel, xoá cache của trình duyệt và truy cập lại trang www.facebook.com. Lúc này, còn truy cập được trang web Facebook không?*



Truy cập facebook.com

Mô tả: Sau khi thực hiện ngắt SSH Tunnel, xoá cache của trình duyệt và truy cập lại trang www.facebook.com. Lúc này, không còn truy cập được trang web Facebook.

➲ *5. Nếu trên Firewall, áp dụng rule chặn kết nối SSH (port 22), lúc này có thể thiết lập tunnel này được hay không? Tại sao?*

```
vulhq@vulhq-virtual-machine:~$ ssh -D 9000 -C vulhq@192.168.103.11
ssh: connect to host 192.168.103.11 port 22: Connection refused
```

Thiết lập turnel

Lúc này không thể thiết lập tunnel.
Vì trên firewall đã áp dụng rule chặn kết nối SSH.

➲ *6. Đề xuất giải pháp để phát hiện và ngăn chặn các cách thức vượt qua sự kiểm soát của Firewall trong trường hợp trên.*

- Hạn Chế Kết Nối SSH Tunneling: Thiết lập tường lửa để chỉ cho phép kết nối SSH từ các máy chủ và địa chỉ IP được xác thực. Điều này giúp hạn chế nguy cơ sử dụng SSH tunneling từ các nguồn không an toàn.

- Kiểm Tra Log SSH: Theo dõi log của SSH để theo dõi bất kỳ hoạt động nào liên quan đến tunneling. Các sự kiện như thiết lập kênh chuyển tiếp có thể được ghi lại và giám sát.
- Giám Sát Kết Nối Telnet: Theo dõi log và giám sát các kết nối Telnet đến các cổng không an toàn. Các sự kiện này có thể là dấu hiệu của hoạt động tunneling.

➲ *7. Đoạn chương trình script.pl trên hoạt động như thế nào?*

```
#!/usr/bin/perl -w
use strict;
use warnings;
```

- *pragma*: thường được sử dụng để thay đổi cách một phần hoặc cả một chương trình Perl được xử lí và biên dịch. Pragma không phải là một thành phần của chương trình mà chỉ là một cấu trúc đặc biệt mà chương trình Perl hiểu được.
- *strict*: cú pháp trong code sẽ được perl kiểm tra nghiêm ngặt hơn. Chương trình sẽ yêu cầu người dùng khai báo đủ tất cả các biến trước khi có thể sử dụng chúng và yêu cầu thêm các nguyên tắc nghiêm ngặt khác. Điều này giúp chương trình tránh được nhiều lỗi và giữ cho source code Perl của bạn clean hơn và dễ hiểu hơn.

```
select STDOUT; $| = 1;
```

- Tiêu chuẩn đầu ra là STDOUT.
- $| = 1: câu lệnh này thiết lập chức năng ghi không đệm, khi $I được đặt thành 1 chế độ ghi không đệm sẽ được bật, khi đó dữ liệu sẽ được ghi ra lập tức mà không cần đợi buffer đầy hoặc có lệnh ‘flush’ được gọi.

```
while (<>)
{
 my @parts = split;
 my $url = $parts[0];
 if ($url =~ /example\.com/)
 {
  # URL Rewriting
  print "http://www.uit.edu.vn\n";
 }
 else
 {
  # No Rewriting.
  print "\n";
 }
}
```

- While (<>) : Khởi tạo vòng lặp đọc từng dòng đầu vào cho tới khi hết đầu vào.
- My @parts = split: chia nhỏ dữ liệu đưa vào mảng parts mỗi khi có khoảng trắng.

- My $url = $parts[0] ; : lấy phần tử đầu tiên của mảng (thường là link url).
- Dòng lệnh if-else: xét điều kiện nếu url là example.com -> chuyển thành http://www.uit.edu.vn.

➲ *8. Thay đổi nội dung đoạn chương trình trên để khi truy cập vào website example.com, một hình ảnh cảnh báo dừng lại xuất hiện (như hình dưới).*

```
 GNU nano 4.8                                                    /et
#!/usr/bin/perl -w
use strict;
use warnings;
# Forces a flush after every write or print on the STDOUT
select STDOUT; $| = 1;
# Get the input line by line from the standard input.
# Each line contains an URL and some other information.
while (<>)
{
my @parts = split;
my $url = $parts[0];
if ($url =~ /example\.com/)
{
# URL Rewriting
print "https://en.wikipedia.org/wiki/File:Soviet_Red_Army_Hammer_and_Sickle.svg\n"
}
else
{
# No Rewriting.
print "\n";
}
}
```

Code

➲ *9. Thay đổi nội dung chương trình để khi truy cập website, tất cả các hình ảnh đều được thay bằng hình ảnh bạn thích (như hình minh hoạ dưới).*

➲ *10. Firewall pfSense hỗ trợ các giao thức thiết lập kết nối VPN nào? Những giao thức này có đặc điểm gì khác nhau?*

Firewall pfSense hỗ trợ các giao thức thiết lập kết nối VPN sau:

- IPsec
- L2TP
- OpenVPN

Sự khác nhau giữa cái giao thức thiết lập này:

	IPsec	L2TP	OpenVPN

Loại giao thức	Giao thức tích hợp vào lớp mạng IP	Giao thức Layer 2 Forwarding.	Giao thức ứng dụng hoạt động qua UDP hoặc TCP
Mức độ bảo mật	Cao	Trung bình	Cao
Mã hóa dữ liệu	Cung cấp các thuật toán mã hóa mạnh mẽ	Cần được kết hợp với IPsec để mã hóa	Sử dụng SSL/TLS để mã hóa dữ liệu
Khả năng tương thích	Tương thích trên nhiều hệ điều hành và thiết bị.	Tương thích trên nhiều hệ điều hành và thiết bị, nhưng cần phần mềm bổ sung để chạy trên Windows	Tương thích rộng rãi, cả trên nền tảng hệ điều hành và các thiết bị di động

➲ *11. Tìm hiểu và thực hiện cấu hình trên pfSense, sao cho từ máy VM B có thể mở kết nối VPN đến pfSense server để truy cập được máy VM A.*