Khung đồ án

I. Cơ sở lý thuyết:

A) Tìm hiểu nguyên lí hoạt động của firewall:

Firewall là một hệ thống an ninh mạng, dựa trên cả phần cứng lẫn phần mềm của thiết bị, sử dụng một bộ các quy tắc điểm kiểm soát traffic vào ra trong khi máy chủ đang hoạt động trên mạng. Tường lửa hoạt động như một rào chắn giữa mạng an toàn và mạng không an toàn. Nó kiểm soát các truy cập đến nguồn lực của mạng thông qua một mô hình kiểm soát chủ động. Nghĩa là, chỉ những traffic phù hợp với chính sách (policy) được định nghĩa trong tường lửa mới được truy cập vào mạng, mọi traffic khác đều bị từ chối.

B) Lý do cần firewal đối với các máy được nối mạng:

Mỗi máy tính trực tuyến lại có một chữ ký điện tử riêng, được gọi là Internet Protocol address (hay còn gọi là địa chỉ IP): Nếu không có firewall hỗ trợ, nó chẳng khác gì chuyện bạn bật tất cả đèn lên và mở rộng cửa để đón trộm vào.

C) Các loại Firewall hiện nay:

1. Personal firewall:

Loại này được thiết kế để bảo vệ một máy tính trước sự truy cập trái phép từ bên ngoài. Bên cạnh đó thì Personal Firewall còn được tích hợp thêm tính năng như theo dõi các phần mềm chống virus, phần mềm chống xâm nhập để bảo vệ dữ liệu. Một số Personal Firewall thông dụng như: Microsoft Internet connection firewall, Symantec personal firewall, Cisco Security Agent…. Loại Firewall này thì thích hợp với cá nhân bởi vì thông thường họ chỉ cần bảo vệ máy tính của họ, thường được tích hợp sẵn trong máy tính Laptop, máy tính PC..

1. Network firewall:

Được thiết kế ra để bảo vệ các host trong mạng trước sự tấn công từ bên ngoài. Chúng ta có các Appliance-Based network Firewalls như Cisco PIX, Cisco ASA, Juniper NetScreen firewall, Nokia firewalls, Symantec’s Enterprise Firewall. Hoặc một số ví dụ về Software-Base firewalls include Check Point’s Firewall, Microsoft ISA Server, Linux-based IPTables.

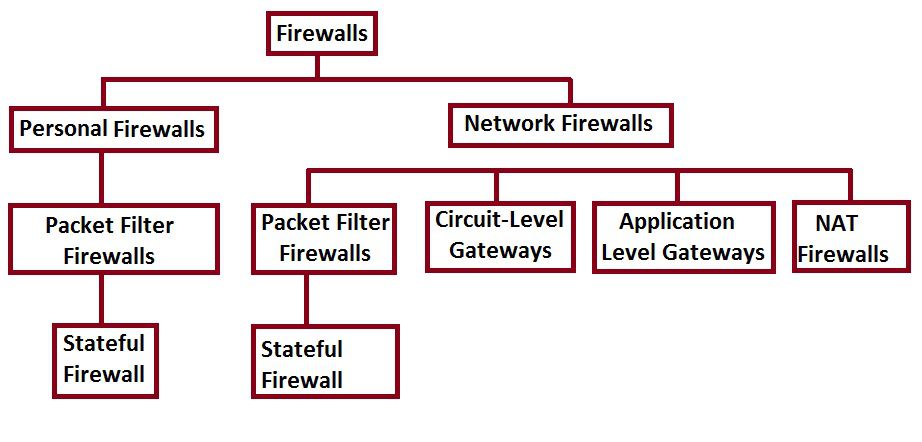
Điểm khác nhau giữa 2 loại firewall là Personal chỉ bảo vệ cho một máy tính duy nhất, còn Network sẽ bảo vệ toàn cục cho các máy ở trong mạng đó.

Trong đó, hệ thống Network Firewall được cấu tạo bởi các thành phần chính như sau:

Bộ lọc Packet (Packet- Filtering Router)

Cổng ứng dụng ( đó là Application-Level Gateway hay Proxy Server).

Cổng mạch (Circuite Level Gateway)



D) Nhiệm vụ chính của firewall:

Firewall hỗ trợ máy tính kiểm soát luồng thông tin giữa intranet và internet, Firewall sẽ quyết định dịch vụ nào từ bên trong được phép truy cập ra bên ngoài, những người nào bên ngoài được phép truy cập vào bên trong hệ thống, hay là giới hạn truy cập những dịch vụ bên ngoài của những người bên trong hệ thống, mình lấy ví dụ như giới hạn trang Facebook, tất cả những người trong hệ thống sẽ không thể truy cập vào được mạng xã hội này. Sau đây là một số nhiệm vụ chính của Firewall:

* Cho phép hoặc vô hiệu hóa các dịch vụ truy cập ra bên ngoài, đảm bảo thông tin chỉ có trong mạng nội bộ.
* Cho phép hoặc vô hiệu hóa các dịch vụ bên ngoài truy cập vào trong.
* Phát hiện và ngăn chặn các cuộc tấn công từ bên ngoài.
* Hỗ trợ kiểm soát địa chỉ truy cập (bạn có thể đặt lệnh cấm hoặc là cho phép).
* Kiểm soát truy cập của người dùng.
* Quản lý và kiểm soát luồng dữ liệu trên mạng.
* Xác thực quyền truy cập.
* Hỗ trợ kiểm soát nội dung thông tin và gói tin lưu chuyển trên hệ thống mạng.
* Lọc các gói tin dựa vào địa chỉ nguồn, địa chỉ đích và số Port ( hay còn cổng), giao thức mạng.
* Người quản trị có thể biết được kẻ nào đang cố gắng để truy cập vào hệ thống mạng.
* Firewall hoạt động như một Proxy trung gian.
* Bảo vệ tài nguyên của hệ thống bởi các mối đe dọa bảo mật.
* Cân bằng tải: Bạn có thể sử dụng nhiều đường truyền internet cùng một lúc, việc chia tải sẽ giúp đường truyền internet ổn định hơn rất nhiều.
* Tính năng lọc ứng dụng cho phép ngăn chặn một số ứng dụng mà bạn muốn. Ví dụ như Facebook Messenger, Skype, Zalo…

E) Những nhược điểm và hạn chế của Firewall

Không cái gì là toàn diện cả, tuy Firewall cung cấp nhiều tính năng hữu ích để bảo vệ người dùng, song nó vẫn có những nhược điểm như:

+ Firewall không thể bảo vệ các mối nguy hiểm từ bên trong nội bộ. Tác hại thì khỏi cần nói các bạn cũng đã biết, nếu một ai trong công ty có ý đồ xấu, muốn phá hoại thì Firewall cũng đành bó tay.

+ Firewall không có đủ thông minh để có thể đọc và hiểu từng loại thông tin và tất nhiên là nó không thể biết được đâu là nội dung tốt và đâu là nội dung xấu. Mà đơn thuần Firewall chỉ hỗ trợ chúng ta ngăn chặn sự xâm nhập của những nguồn thông tin không mong muốn như­ng phải xác định rõ các thông số địa chỉ.

+ Firewall không thể ngăn chặn các cuộc tấn công nếu như cuộc tấn công đó không “đi qua” nó. Ví dụ cụ thể đó là Firewall không thể chống lại một cuộc tấn công từ một đư­ờng dial-up, hoặc là sự dò rỉ thông tin do dữ liệu bị sao chép bất hợp pháp ra đĩa mềm.

+ Firewall cũng không thể chống lại các cuộc tấn công bằng dữ liệu (data-drivent attack). Khi có một số ứng dụng hay phần mềm.. đ­ược chuyển qua th­ư điện tử (ví dụ như Gmail, Yahoo mail…), nó có thể v­ượt qua Firewall vào trong mạng được bảo vệ.

+ Firewall không thể làm nhiệm vụ rà quét virus trên các dữ liệu đ­ược chuyển qua nó, do tốc độ làm việc, sự xuất hiện liên tục của các virus mới và do có rất nhiều cách để mã hóa dữ liệu để có thể thoát khỏi khả năng kiểm soát của firewall. Tuy nhiên, chúng ta không thể phủ nhận một điều rằng Firewall vẫn là giải pháp hữu hiệu đư­ợc áp dụng khá rộng rãi hiện nay.

IPTABLES và cài đặt Firewall trong Ubuntu

1. iptables là gì?

Iptables Linux firewall được sử dụng để theo dõi lưu lượng truy cập đến và đi ở một máy chủ và lọc nó dựa trên các rules do người dùng định nghĩa trước đó để ngăn chặn bất cứ ai truy cập vào hệ thống. Sử dụng Iptables, bạn có thể định nghĩa các rules chỉ cho phép lưu lượng được chọn lọc trên máy chủ của bạn.

1. Khái niệm về các quy tắc trong iptables

Tất cả các dữ liệu được gửi đi trong các gói tin được định dạng qua internet. Linux kernel cung cấp một giao diện để lọc cả các gói tin đi vào và ra sử dụng một bảng các bộ lọc gói tin. Iptables là một ứng dụng dòng lệnh và là một bức tường lửa Linux mà bạn có thể sử dụng để thiết lập, duy trì và kiểm tra các bảng này. Bạn có thể thiết lập nhiều bảng khác nhau, mỗi bảng có thể chứa nhiều chuỗi, mỗi một chuỗi là một bộ quy tắc. Mỗi quy tắc định nghĩa phải làm gì với gói tin nếu nó phù hợp với gói đó. Khi một gói tin được xác định, nó sẽ đưa ra một TARGET. Một target (mục tiêu) có thể là một chuỗi khác để khớp với một trong các giá trị đặc biệt sau đây:

* ACCEPT : gói tin sẽ được phép đi qua.
* DROP: gói tin sẽ không được phép đi qua.
* RETURN: bỏ qua chuỗi hiện tại và quay trở lại quy tắc tiếp theo từ chuỗi mà nó được gọi.

Cấu hình iptables trong Linux-Ubuntu:

1. Kiểm tra trạng thái hiện tại của iptables:

sudo iptables -L -v

Đầu ra :

Chain INPUT (policy ACCEPT 309K packets, 416M bytes)

pkts bytes target prot opt in out source destination

Chain FORWARD (policy DROP 0 packets, 0 bytes)

pkts bytes target prot opt in out source destination

0 0 DOCKER-ISOLATION all -- any any anywhere anywhere

0 0 DOCKER all -- any docker0 anywhere anywhere

0 0 ACCEPT all -- any docker0 anywhere anywhere ctstate RELATED,ESTABLISHED

0 0 ACCEPT all -- docker0 !docker0 anywhere anywhere

0 0 ACCEPT all -- docker0 docker0 anywhere anywhere

Chain OUTPUT (policy ACCEPT 167K packets, 22M bytes)

pkts bytes target prot opt in out source destination

Chain DOCKER (1 references)

pkts bytes target prot opt in out source destination

Chain DOCKER-ISOLATION (1 references)

pkts bytes target prot opt in out source destination

0 0 RETURN all -- any any anywhere anywhere

Giải thích:

Tùy chọn -L để liệt kê tất cả các rules (qui tắc) và -v để hiện thêm các danh sach bổ trợ.

1. Định nghĩa các chain rules:

Bảng các Argurments string

| Prefixs | Features |
| --- | --- |
| -A,--append | Add one or more new rules to the specified chain |
| -D,--delete | Delete one or more rules from specified chain |
| -P, --policy | Set policy of designated chain to specified target |
| -N, --new-chain | Create a new user-defined chain |
| -X, --delete-chain | Delete specified user-defined chain |
| -F | Table initialization |
| -p, --protocol | Specify protocol protocol (tcp, udp, icmp, all) |
| -s, - source IP address [/ mask] | Source address. Describe IP address or host name |
| -d, - destination IP address [/ mask] | Address of the destination. Describe IP address or host name |
| -i , - in - interface | Specifies the interface on which the device packet comes in |
| -o, - out - interface | Specify the interface on which the device packet appears |
| -j, --jump | Specify action when matching target condition |
| -t, --table | Specify table table |
| -m state - state state | Specify condition of packet as condition |
| ! | Invert the condition (except for ~) |
| State can be NEW, ESTABLISHED, RELATED, INVALID |  |

Định nghĩa các chain rules là thực hiện thêm nó vào danh sách chain hiện tại. Đây là lệnh Iptables được định dạng với các tùy chọn thông thường:

sudo iptables -A -i <interface> -p <protocol (tcp/udp)> -s <source> --dport <port no.> -j <target>

-A: thêm chain rules

-i <interface> là giao diện mạng bạn cần thực hiện lọc các gói tin

-p <protocol> là giao thức mạng thực hiện lọc (tcp/udp)

–dport <port no.> là cổng mà bạn muốn đặt bộ lọc

Thực hiện khai báo các cổng dịch vụ cần bảo vệ:

SSH=22

FTP=20,21

DNS=53

SMTP=25,465,587

POP3=110,995

IMAP=143,993

HTTP=80,443

IDENT=113

NTP=123

MYSQL=3306

NET\_BIOS=135,137,138,139,445

DHCP=67,68

1. Các bước thực hiện cài đặt shell script để init iptables:

Khi khởi tạo iptables, cần lưu ý xoá hết tất cả các chain rules, các biến đếm, và các accept rules cơ bản

Vd hàm initialize():

initialize()

{

iptables -F # Table initialization

iptables -X # Delete chain

iptables -Z # Clear packet counter and byte counter

iptables -P INPUT ACCEPT

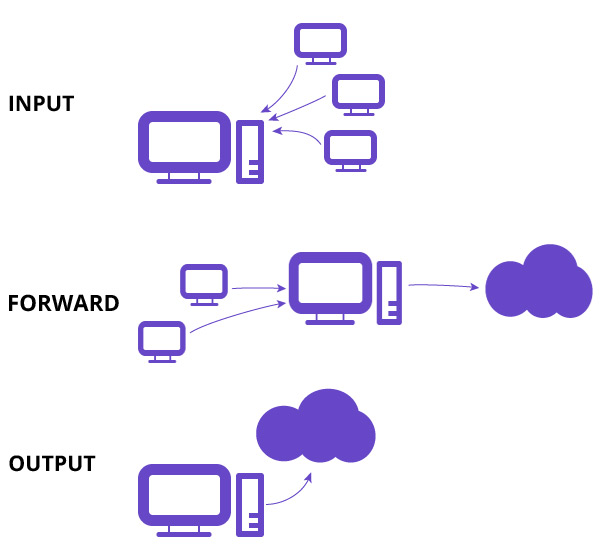
iptables -P OUTPUT ACCEPT

iptables -P FORWARD ACCEPT

}

Lí giải về INPUT, OUTPUT và FORWARD:

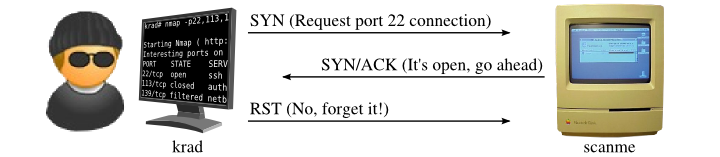
* INPUT – được sử dụng để điều khiển các gói tin đến tới máy chủ. Bạn có thể chặn hoặc cho phép kết nối dựa trên cổng, giao thức hoặc địa chỉ IP nguồn.
* FORWARD – được sử dụng để lọc các gói dữ liệu đến máy chủ nhưng sẽ được chuyển tiếp ở một nơi khác.
* OUTPUT – được sử dụng để lọc các gói tin đi ra từ máy chủ của bạn.



Stealth scan/ TCP SYN (Stealth) Scan (-sS)

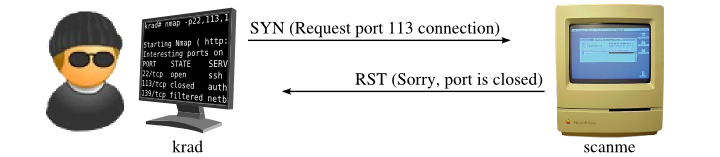
Quét SYN là tùy chọn quét mặc định và phổ biến nhất vì nhiều lý do (hầu hết là chính đáng). Nó có thể được thực hiện nhanh chóng, quét hàng nghìn cổng mỗi giây trên một mạng mà không bị cản trở bởi các tường lửa intrusive.

Quét SYN tương đối kín đáo và lén lút vì nó không bao giờ hoàn thành kết nối TCP (theo quy tắc bắt tay 3 bước). SYN scan quét ở cấp độ packet, cùng xem ví dụ tấn công vào port 22 như bên dưới:



Nmap bắt đầu gửi một packet TCP với cờ SYN đã được set (tham chiếu đến ip header) và gửi đến port 22. Đây là bước đầu tiên trong quá trình bắt tay 3 bước mà mọi kết nối hợp pháp đều phải trải qua. Khi cổng mục tiêu (22) mở, scanme thực hiện bước thứ 2 bằng cách gửi phản hồi cờ SYN và ACK. Trong một kết nối bình thường, krad phải hoàn thành bước thứ 3 bằng việc gửi một packet ACK chấp nhận SYN/ACK. Nmap thì không cần phải làm vậy, vì phản hồi SYN/ACK đã phản hồi rằng port đã mở. Nếu Nmap hoàn thành được kết nối, sau đó nó phải chịu trách nhiệm đóng lại. Việc này yêu cầu một quy trình bắt tay mới, sử dụng packet FIN thay vì SYN. Vậy ACK không phải là một ý tưởng tốt, nhưng vẫn cần phải sử dụng nó trong một số tình huống. Nếu như SYN/ACK hoàn toàn bị bỏ qua, scanme sẽ giả sử rằng nó đã bị drop và tiếp tục gửi lại. Câu trả lời thích hợp ở tình huống này , vì chúng ta không muốn tạo một kết nối đầy đủ, là một gói RST như được hiển thị trong biểu đồ. Việc này giống như thông báo cho scanme quên kết nối đã thử. Nmap có thể gửi packet RST này một cách dễ dàng, nhưng nó thực sự không cần phải làm như vậy. Hệ điều hành chạy trên krad cũng nhận SYN/ACK, nhưng điều này nằm ngoài mong đợi vì Nmap đã tạo ra đầu dò SYN cho chính nó. Vì vậy, hệ điều hành đáp ứng với các SYN / ACK không mong đợi với một gói RST. Mọi gói RST được mô tả trong phần SYN scan này đều có một ACK bit set vì chúng luôn được gửi trong gói packet phản hồi. Điều này dẫn đến bit đó không được hiển thị rõ ràng cho RST packets bởi vì quá trình bắt tay 3 bước sẽ không bao giờ được hoàn tất. SYN scan đôi lúc còn được biết đến với tên gọi half-open scanning.

Hình dưới mô tả quá trình nmap xác nhận cổng 113 đã đóng. Việc này thậm chí còn dễ dàng hơn trường hợp mở kết nối. Bước đầu tiên khá giống nhau, nmap gửi một xác thực SYN đến scanme. Nhưng thay vì nhận SYN/ACK trở lui, nó trả về một RST. Việc gửi phản hồi RST xem như cổng đã đóng, không cần phải lưu ý đến giao tiếp của cổng này nữa.



Cuối cùng, krad cho chúng thấy làm thế nào mà một port được lọc xuất hiện trong Nmap ở hình .... SYN khơi mào được gửi đầu tiên như thường lệ, nhưng nmap không thấy reply nào. Phản hồi có thể được gửi chậm. Từ phản hồi trước (hoặc theo mức timming quy định sẵn), nmap biết được cần phải đợi bao lâu là tuần tự từ bỏ những packet nhận được. Một cổng không phản hồi thường được lọc qua (thường là chặn bởi firewall của thiết bị, hoặc có thể host đã chết), nhưng (test ....)