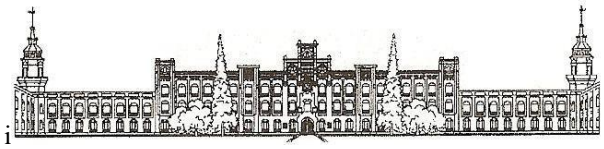

Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут»

Лабораторна робота №7

з дисципліни «Комп'ютерні мережі»

«Пакетні фільтри»

Виконала студентка групи: КВ-11

ПІБ: Михайліченко Софія

Перевірив:_____

Київ 2024

Мета роботи:

Поглиблене самостійне вивчення спеціальних питань, присвячених організації та конфігуруванню брандмауера, що використовує iptables.

План виконання лабораторної роботи:

1. Ознайомитися та засвоїти теоретичні відомості викладені в навчально-методичному посібнику до лабораторної роботи.
2. Виконати завдання до лабораторної роботи. Скласти звіт.

Завдання:

1. Дозволити приймати пакети лише з хоста scs.kpi.ua. Виконайте перевірку дії встановлених правил.
2. Заборонити з’єднання з вашою машиною з комп’ютерів в локальній мережі.
3. Відключити відклик на ICMP запити на вашій машині. Виконайте перевірку дії встановлених правил.
4. Закрити всі порти, крім SMTP. Виконайте перевірку дії встановлених правил.
5. Створити ланцюжок користувача spammsg. В цьому ланцюжку записати правило, за яким усі повідомлення від джерела спаму будуть знищуватися. Виконати перехід з ланцюжка INPUT на ланцюжок spammsg.
6. Показати викладачу.

Теоретичні відомості:

Організація простого брандмауера

Брандмауер (firewall) є важливим елементом захисту комп'ютерних мереж, що служить для контролю і фільтрації трафіку, який проходить між внутрішньою мережею і зовнішнім середовищем. Зазвичай його розміщують на межі мережі, щоб контролювати передачу даних та запобігти несанкціонованому доступу ззовні або всередині мережі. Брандмауери можуть використовуватися як для захисту корпоративних мереж, так і для обмеження доступу користувачів всередині мережі.

Типи брандмауерів

1. Пакетні фільтри – це брандмауери, які фільтрують пакети на основі таких параметрів, як IP-адреси, номери портів і протокол. Вони аналізують заголовки пакетів і приймають рішення щодо їх пропуску або блокування. Пакетні фільтри ефективні для загальної фільтрації, але вони не можуть глибоко аналізувати вміст пакетів, що обмежує їх захисні властивості.
2. Брандмауери прикладного рівня – це брандмауери, які контролюють трафік, орієнтуючись на протоколи високого рівня, такі як HTTP, FTP або SMTP. Вони зазвичай інтегрують функціональність проксі-серверів і здійснюють глибокий аналіз на рівні додатків. Брандмауери цього типу можуть перевіряти автентифікацію користувачів та використовувати більш точні механізми захисту.
3. Брандмауери рівня з’єднання – ці брандмауери створюють з'єднання між відправником і приймачем, використовуючи відповідні порти та протоколи. Вони є більш універсальними, оскільки можуть працювати з великою кількістю протоколів, на відміну від брандмауерів прикладного рівня, що вимагають специфічного програмного забезпечення для кожного сервісу.
4. Брандмауери нового покоління (NGFW) – цей тип брандмауерів поєднує в собі функції фільтрації пакетів, інспекції трафіку, а також інші інструменти захисту, такі як система виявлення вторгнень (IDS), система профілактики вторгнень (IPS), підтримка VPN і глибокий аналіз пакунків (DPI). Брандмауери NGFW можуть здійснювати глибокий аналіз шифрованого трафіку SSL/TLS, що робить їх значно ефективнішими в сучасних умовах кіберзагроз.

Міжмережевий екран Netfilter/iptables

Netfilter є брандмауером, який вбудований в ядро Linux починаючи з версії 2.4. Вони дозволяють виконувати фільтрацію і перенаправлення пакетів на основі різних правил, зокрема за допомогою утиліти **iptables**.

Iptables — це інструмент командного рядка, який дозволяє адміністратору Linux керувати фільтрацією пакетів через системи Netfilter. Вона дає можливість створювати, змінювати та видаляти правила фільтрації трафіку, створювати власні ланцюжки та таблиці. Всі правила фільтрації, що встановлюються через iptables, виконуються на основі певних критеріїв, таких як адреси IP, номери портів і протоколи. В залежності від цих правил, пакети можуть бути прийняті, відкинуті або передані в інший ланцюг для подальшої обробки.

Принцип роботи Netfilter/iptables

Система фільтрації пакетів Netfilter використовує три основні ланцюги: INPUT, OUTPUT і FORWARD. Кожен ланцюг має набір правил, які перевіряють пакети. Пакет може бути:

- Прийнятий (ACCEPT).
- Відхидений (DROP).
- Перенаправлений в інший ланцюг для додаткової перевірки.

В залежності від правил і політики ланцюга, що застосовуються до пакету, він може бути переданий в наступний ланцюг або відкинутий.

Також Netfilter підтримує таблиці: filter, mangle, nat, raw. Кожна таблиця має свої цілі:

- **filter** — основна таблиця для фільтрації пакетів.
- **mangle** — для модифікації пакунків.
- **nat —** для трансляції адреси.
- **raw —** для спеціальної обробки пакунків перед передачею.

Хід виконання роботи

1. Спочатку за допомогою команди iptables дозволяємо приймати лише пакети з хоста scs.kpi.ua:

```
student@virt-linux:~$ sudo iptables -A INPUT -s scs.kpi.ua -j ACCEPT
[sudo] password for student:
student@virt-linux:~$ sudo iptables -A INPUT -j DROP
student@virt-linux:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  77.47.131.42         anywhere
DROP       all  --  anywhere             anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
student@virt-linux:~$ _
```

Цей результат ілюструє, як команда iptables налаштовує правила для фільтрації мережевого трафіку.

Перша команда додає правило для ланцюга INPUT, яке дозволяє весь вхідний трафік із хоста scs.kpi.ua. При цьому в списку правил з’являється його IP-адреса (наприклад, 77.47.131.42), яка асоціюється з цим доменом. Друга команда додає ще одне правило, яке блокує будь-який інший вхідний трафік, незалежно від його джерела та призначення. Після цього команда iptables -L виводить список активних правил.

Ланцюг INPUT тепер має два правила: одне, яке дозволяє доступ із конкретного IP-адреса, і інше, яке блокує всі інші підключення. Політика за замовчуванням для цього ланцюга все ще встановлена як ACCEPT, але через те, що останнє правило блокує будь-який трафік, всі інші з'єднання, крім дозволених, будуть відхилені.

Ланцюги FORWARD і OUTPUT залишаються з політикою ACCEPT, що означає, що трафік для пересилання і вихідний трафік не обмежені додатковими правилами.

Тепер виконаємо перевірку дії встановлених правил:

```
student@virt-linux:~$ ping scs.kpi.ua
PING scs.kpi.ua (77.47.131.42) 56(84) bytes of data.
64 bytes from scs.kpi.ua (77.47.131.42): icmp_seq=1 ttl=56 time=9.41 ms
64 bytes from scs.kpi.ua (77.47.131.42): icmp_seq=2 ttl=56 time=10.7 ms
64 bytes from scs.kpi.ua (77.47.131.42): icmp_seq=3 ttl=56 time=8.98 ms
64 bytes from scs.kpi.ua (77.47.131.42): icmp_seq=4 ttl=56 time=45.4 ms
64 bytes from scs.kpi.ua (77.47.131.42): icmp_seq=5 ttl=56 time=9.33 ms
64 bytes from scs.kpi.ua (77.47.131.42): icmp_seq=6 ttl=56 time=10.3 ms
64 bytes from scs.kpi.ua (77.47.131.42): icmp_seq=7 ttl=56 time=9.71 ms
64 bytes from scs.kpi.ua (77.47.131.42): icmp_seq=8 ttl=56 time=8.32 ms
64 bytes from scs.kpi.ua (77.47.131.42): icmp_seq=9 ttl=56 time=11.2 ms
64 bytes from scs.kpi.ua (77.47.131.42): icmp_seq=10 ttl=56 time=8.89 ms
^C
--- scs.kpi.ua ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9080ms
rtt min/avg/max/mdev = 8.323/13.220/45.362/10.745 ms
```

Бачимо, що результат виконання команди ping до домену scs.kpi.ua є стабільним. Ця команда перевіряє доступність хоста, відправляючи ICMP-запити і отримуючи відповіді. У виводі видно, що домен успішно резолвиться в IP-адресу 77.47.131.42, що означає, що з DNS проблем немає. Кожен рядок виводу відображає час, за який пакет дійшов до хоста і повернувся назад. Затримка варіюється від 8.32 мс до 45.36 мс, з середнім часом у 13.22 мс. Всі 10 відправлених пакетів були отримані без втрат, що вказує на стабільне з'єднання. Відхилення часу відповіді складає 10.75 мс, що також свідчить про нормальну роботу мережі без великих коливань у затримці.

2. Заборонимо з’єднання з нашою віртуальною машиною з комп’ютерів в локальній мережі:

```
student@virt-linux:~$ sudo iptables -A INPUT -s 192.168.0.0/24 -j DROP
student@virt-linux:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  scs.kpi.ua          anywhere
DROP       all  --  192.168.0.0/24      anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

Тепер виконуємо перевірку дії встановлених правил за допомогою команди **ping** на іншому копм'ютері:

```
Командний рядок
Microsoft Windows [Version 10.0.19045.5131]
(c) Корпорація Майкрософт. Усі права захищені.

C:\Users\Acer>ping 10.0.2.15

Pinging 10.0.2.15 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.0.2.15:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\Users\Acer>
```

3. Тепер можемо відключити виклик на ICMP запити на нашій віртуальній машині:

```
student@virt-linux:~$ sudo iptables -A INPUT -p icmp -j DROP
student@virt-linux:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  scs.kpi.ua           anywhere
DROP       icmp --  anywhere             anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
student@virt-linux:~$
```

Та виконуємо перевірку дій встановлених правил:

```
Командний рядок
Microsoft Windows [Version 10.0.19045.5131]
(c) Корпорація Майкрософт. Усі права захищені.

C:\Users\Acer>ping 10.0.2.15

Pinging 10.0.2.15 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.0.2.15:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\Users\Acer>
```

4. Закриваємо всі порти, крім STMP:

```
student@virt-linux:~$ sudo iptables -F
student@virt-linux:~$ sudo iptables -X
student@virt-linux:~$ sudo iptables -P INPUT ACCEPT
student@virt-linux:~$ sudo iptables -P OUTPUT ACCEPT
student@virt-linux:~$ sudo iptables -P FORWARD ACCEPT
student@virt-linux:~$ sudo iptables -A INPUT -p tcp --dport 25 -j ACCEPT
student@virt-linux:~$ sudo iptables -A INPUT -p tcp -j DROP
student@virt-linux:~$ sudo iptables -L -v
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     tcp  --  any    any     anywhere             anywhere             tcp dpt:smtp
    4   208 DROP       tcp  --  any    any     anywhere             anywhere

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 5 packets, 260 bytes)
 pkts bytes target     prot opt in     out     source               destination
student@virt-linux:~$ sudo nmap -p 1-65535 192.168.0.108
Starting Nmap 7.80 ( https://nmap.org ) at 2024-12-21 22:45 EET
Nmap scan report for virt-linux (192.168.0.108)
Host is up (0.000079s latency).
Not shown: 65534 filtered ports
PORT   STATE SERVICE
25/tcp open  smtp

Nmap done: 1 IP address (1 host up) scanned in 106.53 seconds
```

5. Тепер створимо ланцюжок користувача spammsg. В цьому ланцюжку запишемо правило, за яким усі повідомлення від джерела спаму будуть знищуватися:

```
student@virt-linux:~$ sudo iptables -N spammsg
```

Створення заданого ланцюжка spammsg.

```
student@virt-linux:~$ sudo iptables -A spammsg -s 192.168.0.103 -j DROP
```

Встановлення відповідного правила для видалення повідомлень від джерела спаму

```
student@virt-linux:~$ sudo iptables -A INPUT -j spammsg
```

Тепер додаємо правило для перенаправлення трафіку на ланцюжок "spammsg"

Та тепер можемо переглянути встановлені правила

```
student@virt-linux:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
spammsg    all  --  anywhere             anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain spammsg (1 references)
target     prot opt source               destination
DROP       all  --  192.168.0.103        anywhere
```

І перевіряємо це правило на іншому ПК

```
Командний рядок
Microsoft Windows [Version 10.0.19045.5131]
(c) Корпорація Майкрософт. Усі права захищені.

C:\Users\Acer>ping 10.0.2.15

Pinging 10.0.2.15 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.0.2.15:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\Users\Acer>
```

Як бачимо, всі правила спрацьовують, а отже все працює вірно.

Висновок

У цій лабораторній роботі було детально досліджено конфігурацію та налаштування брандмауера за допомогою інструменту iptables. Реалізовано низку завдань, які допомогли набути базових навичок адміністрування мережевого доступу в Linux за допомогою фільтрації трафіку.
Завдяки iptables було успішно заблоковано отримання пакетів з інших джерел, окрім вказаного домену. Перевірка показала, що підключення через IP-адресу працює коректно.
Застосування правил фільтрації на брандмауері призвело до блокування з'єднань з іншими комп'ютерами в локальній мережі, що підтвердило ефективність налаштувань у обмеженні таких з'єднань.
Інструмент iptables також використано для відключення відповідей на ICMP запити (ping), що підтвердило відсутність реакції системи на пінгування.
Порти, що не використовуються, були закриті, а доступ до порту SMTP (25) залишено відкритим. Перевірка відкритих портів через інструмент nc (Netcat) підтвердила коректність налаштувань.
Завдяки iptables було створено ланцюг spammsg, в якому прописано правило для блокування спам-повідомлень з вказаних джерел, і перевірка показала, що ці повідомлення були успішно заблоковані.