Міністерство освіти і науки України

Тернопільський національний педагогічний університет
імені Володимира Гнатюка

Кафедра інформатики та
методики її викладання

Фільтрація трафіку брандмауером iptables з врахуванням стану TCP-пакетів

Підготував:
студент групи І-24
фізико-математичного факультету
Задорожний Олесь
Науковий керівник:
Олексюк В.П.

Тернопіль – 2013

Зміст

Вступ

Дана тема є надзвичайно актуальною в сфері комп'ютерних мереж, адже без фільтрації важко обходитись в даний час.

Брандмауер – одна із основних ліній захисту будь-якого сервера, і саме від правильності його настроювання залежатиме чи зможе зловмисник просунутися далі у своїх спробах проникнення в систему.

Головним завданням всіх зломщиків – отримання доступу до командного інтерпретатора сервера для використання його можливостей у своїх інтересах. Найчастіше проникнення здійснюються за допомогою дір у сервісах або через підбір пароля.

З iptables можна робити безліч речей. Однією із основних – це керування цілими ланцюжками.

Основною метою даного завдання – створення правил для фільтрації пакетів, вивчення базового набору ключів програми iptables і отримання практичних навичок щодо налаштування пакетного фільтра. Для роботи ми будемо використовувати ОС Linux і звичайно утиліту для фільтрації – iptables.

В теоретичній частині буде описуватись інформація про саму утиліту, її синтаксис, принцип роботи, важливість при роботі тощо.

В практичній частині буде показано ряд правил щодо застосування даної утиліти і як це впливає на роботу комп'ютера. Також до кожного правила буде створено опис, в якому буде пояснюватись його функція і завдання.

1. Теоретична частина

Iptables — утиліта командного рядка, стандартний інтерфейс керування роботою міжмережевного екрану (брандмауеру) Netfilter для ядер Linux від версії 2.4. Всупереч поширеній думці, ані iptables, ані netfilter не виконують маршрутизацію пакетів і не керують нею. Netfilter лише фільтрує та модифікує (також для NAT) пакети за правилами, вказаними адміністратором через утиліту iptables. Для використання утиліти iptables потрібні привілеї суперкористувача (root).

1.1. Завдання брандмауера.

Основним завдання брандмауера є фільтрація та обробка пакетів, що проходять через мережу. При аналізі вхідного пакету він приймає рішення про долю цього пакету: викинути пакет (DROP), прийняти пакет (ACCEPT) або зробити з ним ще щось.

1.2. Можливості iptables.

До основних можливостей iptables відноситься:

1) фільтрація трафіку на основі адрес відправника і одержувача пакетів, номерів портів;
2) перенаправлення пакетів за певними параметрами;
3) організація доступу в мережу (SNAT);
4) обмеження числа підключень;
5) встановлення квот трафіку;
6) виконання правил за розкладом;

1.3. Терміни.

DNAT — від анг. Destination Network Address Translation — зміна мережевої адреси отримувача. DNAT — це зміна адреси призначення у заголовку пакета. Найчастіше використовують у парі з SNAT. Основне застосування — використання єдиної реальної IP-адреси кількома

комп'ютерами для виходу до Інтернету та умов надання додаткових мережевих послуг зовнішнім клієнтам.

Потік (Stream) — під цим терміном мається на увазі з'єднання, крізь яке передаються і приймаються пакети. Я використав цей термін для позначення з'єднання, якими передається меншою мірою 2 пакета в обох напрямах.

SNAT — від анг. Source Network Address Translation — зміна мережевого адресу відправника. SNAT — це й зміна вихідного адресу в заголовку пакета. Основне застосування — використання єдиного реального IP-адресу кількома комп'ютерами для виходу до Інтернету.

Стан (State) — під цим терміном мається на увазі стан, де знаходиться пакет.

Простір користувача (User space) — під цим терміном маю на увазі усе, що розміщено поза ядром, наприклад: команда iptables -h виконується поза ядром, тоді як команда iptables -A FORWARD -p tcp -j ACCEPT виконується (частково) у просторі ядра, оскільки вона додає нове правило до наявного набору.

Простір ядра (Kernel space) — більшою або меншою мірою є твердженням, зворотним терміну «Простір користувача». Має на увазі місце виконання — в середині ядра.

1.4. Принцип роботи iptables.

Всі пакети проходять через певні послідовності ланцюжків. При проходженні пакетом ланцюжка, до нього послідовно застосовуються всі правила цього ланцюжка в порядку їх слідування. Під застосуванням правила розуміється: по-перше, перевірка пакету на відповідність критерію, і по-друге, якщо пакет цьому критерію відповідає, застосування до нього зазначеної дії..

Якщо пакет пройшов через весь базовий ланцюжок і до нього так і не було застосовано жодного термінального дії, до нього застосовується дія за замовчуванням для даної ланцюжка.

Наприклад, якщо наш би комп'ютер збирався надіслати пакет www.yahoo.com із запитом HTML-сторінки, то пакету спершу треба би було пройти через вихідний ланцюг OUTPUT. Ядро перегляне правила в цьому ланцюзі, щоб побачити, чи не збігається пакет із якимось. Перше ж правило, з яким відбувся збіг, вирішує долю пакета. Якщо жодне правило не зійшлося, тоді остаточне рішення визначатиметься політикою цілого ланцюга. Потім відповідь, надіслана нам назад Yahoo!, повинна пройти через ланцюг INPUT (входу).

Одна з найновіших можливостей фільтрації пакетів, реалізованих у системі Linux, дозволяє враховувати при перевірці пакетів стан з'єднання. Засоби дозволяли обробляти окремі пакети , незалежно від того, чи були вони частиною з'єднання. (Раніше вже зустрічалася опція – syn, що дозволяє визначити пакет, що містить запит на встановлення з'єднання. Існують засоби, які надають можливість включити свої пакети в набір пакетів, переданих в рамках діючого з'єднання. Таке включення пакетів називається перехопленням TCP -з'єднання). Засоби перевірки пакетів з урахуванням стану визначають приналежність пакетів до поточного з'єднання , аналізуючи послідовні номери, IP- адреси, зазначені в заголовках , і інші характеристики пакетів. Правила, що реалізують таку перевірку, дозволяють відкидати сторонні пакети, включені до складу даних, які передаються в рамках існуючого з'єднання.

Для включення засобів перевірки пакетів з урахуванням стану використовується опція -state, передує опцією -t стан. Для опції -state можна задати одне або кілька значень. Якщо ви вказуєте декілька значень, вони повинні розділятися комами.

INVALID. Перевірка показала, що пакет не належить відомому з'єднанню і може виявитись фальсифікованим.

NEW. Пакет намагається встановити нове з’єднання.

ESTABLISHED. Пакет відповідає існуючому з’єднанню.

RELATED. Пакет не є частиною існуючого з'єднання, але його присутність припустимо (наприклад, це може бути ICMP-пакет, що повідомляє про помилку).

1.5. Основні команди.

Таблиця 1.1

Синтаксис iptables:

Команда	Приклад	Опис
- A , - append	iptables - A INPUT ...	Додає нове правило в кінець заданої ланцюжка
- D , - delete	iptables - D INPUT - dport 80- j DROP , iptables - D INPUT 1	Видалення правила з ланцюжка .
- R , - replace	iptables - R INPUT 1- s 192.168.0.1 - j DROP	Ця команда замінює одне правило іншим. В основному вона використовується під час налагодження нових правил.
- I , - insert	iptables - I INPUT 1 - dport 80- j ACCEPT	Добавляє нове правило в ланцюжок.
-L, - list	iptables - L INPUT	Виведення списку правил в заданій ланцюжку , в даному прикладі передбачається висновок правил з ланцюжка INPUT .
- F , - flush	iptables - F INPUT	Скидання (видалення) всіх правил із заданої ланцюжка (таблиці) .
- Z , - zero	iptables - Z INPUT	Обнулення всіх лічильників в заданій ланцюжку
- N, - new - chain	iptables - N allowed	Створюється нова ланцюжок з заданим ім'ям в заданій таблиці
- X, - delete - chain	iptables - X allowed	Видалення заданої ланцюжка із заданої таблиці.
- P, - policy	iptables - P INPUT DROP	Задає політику по-замовчуванню для заданої ланцюжка.

1.6. Структура TCP-пакета.

Дані передаються у вигляді пакетів. Така організація передачі означає, що дані, якого розміру вони б не були, розбиваються на окремі фрагменти, які формуються у пакети (формування пакетів передбачає, що до даних додається службовий заголовок), після чого у вигляді пакетів дані передаються по мережі (причому порядок передачі пакетів може порушуватися). Приймаюча система "збирає" з пакетів вихідний масив даних на підставі заголовків пакетів.

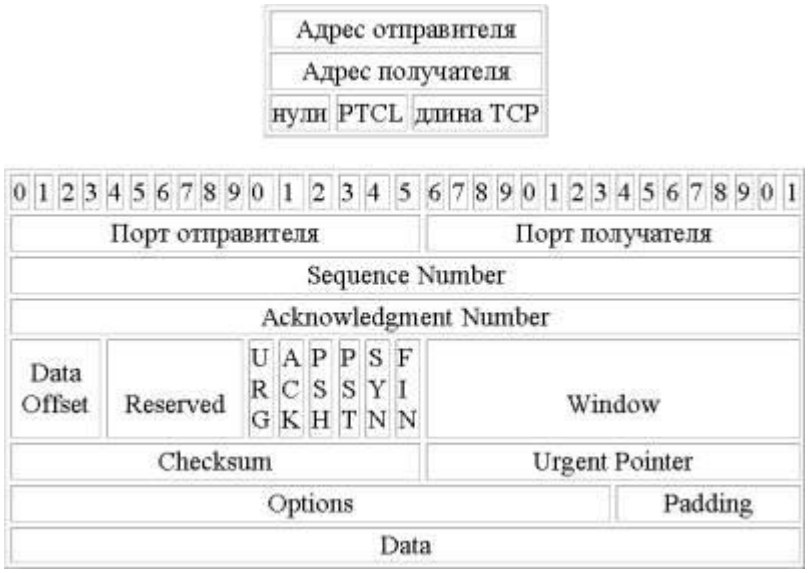


Рис. 1.1

Основні складові: (Рис 1.1)

Sequence Number (SYN) – номер черги або послідовний номер, показує порядковий номер пакету при передачі , саме тому приймаюча система збирає пакети саме так, як треба, а не в тому порядку, як вони прийшли.

Acknowledgment Number (ACK) – номер підтвердження, показує , на пакет з яким SYN відповідає дистанційна система, таким чином ми маємо уявлення, що дистанційна система отримала наш пакет з даними SYN.

Контрольні біти – 6 біт (на схемі між reversed і window).

Значення бітів:

URG:поле термінового показника задіяно;

ACK:поле підтвердження задіяно;

PSH:функція проштовхування;

RST:перезавантаження даного з'єднання;

SYN:синхронізація номерів черги;

FIN:немає більше даних для передачі;

DATA – це безпосередньо ті дані, які ми хочемо передати.

Коли ми хочемо встановити з'єднання, ми відправляємо віддаленій системі пакет наступної структури:

Client --- SYN (856779) --- Host, де Client – це ми, а Host – це віддалена система. Ми відправляємо пакет лише з вказівкою SYN. Це означає, що цей пакет перший , ми ні на що не відповідаємо (відсутній ACK).

Цікавий момент в тому, звідки береться SYN. SYN утворюється від первісного номера черги (ISN) – це 32-бітний номер від 1 до 4294967295 (2 в 32-ому степені). ISN при перезавантаженні системи дорівнює 1, потім кожну секунду він збільшується на 128000 (строго кажучи зміна відбувається кожні 4 мікросекунди) + при кожному встановленому з'єднанні він збільшується на 64000. Виходить, що цикл унікальності ISN, за умови того, що ніякі з'єднання не встановлювалися, становить приблизно 4,55 години. Оскільки жоден пакет так довго по мережі не подорожує, ми можемо вважати, що SYN буде абсолютно унікальним.

Отримавши наш пакет, дистанційна система відповідає, що отримала і готова встановити з'єднання.

Дані пакет виглядає так:

Host --- SYN (758684758) і ACK (856780) --- Client

1.7. TCP-з’єднання

TCP-з'єднання завжди встановлюється передачею трьох пакетів, які ініціюють і встановлюють з'єднання, через яке в подальшому передаватимуться дані. Сесія починається з передачі SYN пакета, у відповідь на який передається SYN/ACK пакет і підтверджує встановлення з'єднання пакет ACK. Після цього з'єднання вважається встановленим і готовим до передачі даних. Для всіх типів з'єднань, трасування проходить практично однаково. Трасувальник, з точки зору користувача, фактично не стежить за ходом встановлення з'єднання.

Просто, як тільки трасувальник "побачив" перший (SYN) пакет, то привласнює йому статус NEW. Як тільки через трасувальника проходить другий пакет (SYN/ACK), то з'єднанню присвоюється статус ESTABLISHED. Будуючи свій набір правил, ви можете дозволити залишати локальну мережу пакетам зі статусом NEW і ESTABLISHED, а у вхідному трафіку пропускати пакети тільки зі статусом ESTABLISHED і все буде працювати прекрасно. І навпаки, якби трасувальник продовжував вважати з'єднання як NEW, то фактично нам ніколи не вдалося б встановити з'єднання з "зовнішнім світом", або довелося б дозволити проходження NEW пакетів в локальну мережу. З точки зору ядра все виглядає більш складним, оскільки в просторі ядра TCP з'єднання мають ряд проміжних станів, недоступних в просторі користувача.

1.8. TCP критерії.

Критерій: -sport,-source-port

Приклад: iptables-A INPUT-p tcp-sport 22

Опис: Вихідний порт, з якого був відправлений пакет.

Критерій: -dport,-destination-port

Приклад: iptables-A INPUT-p tcp-dport 22

Опис: Порт або діапазон портів, на який адресований пакет.

Критерій: -tcp-flags

Приклад: iptables-p tcp-tcp-flags SYN, FIN, ACK SYN

Опис: Визначає маску і прапори tcp-пакета.

Критерій: -tcp-option

Приклад: iptables-p tcp-tcp-option 16

Опис: задовольняє умовам даного критерію буде вважатиметься пакет, TCP параметр якого дорівнює заданому числу.

2. Практична частина

У практичній частині ми будемо працювати зі створенням ряду правил для безпечності комп’ютера з врахуванням стану TCP-пакетів. Тобто, будемо виконувати певні дії, які перевірятиме ядро при виконанні. При цьому нам потрібна буде утиліта iptables із перевіркою стану -m state.

Для початку згадаємо декілька простих правил для дозволу або блокування служб, портів тощо.

2.1. Блокування портів.

Заблокує всі вхідні з'єднання з привілейованих портів (привілейованими в TCP / UDP вважаються порти з 0 по 1023 включно, так як для їх використання потрібні повноваження суперкористувача).

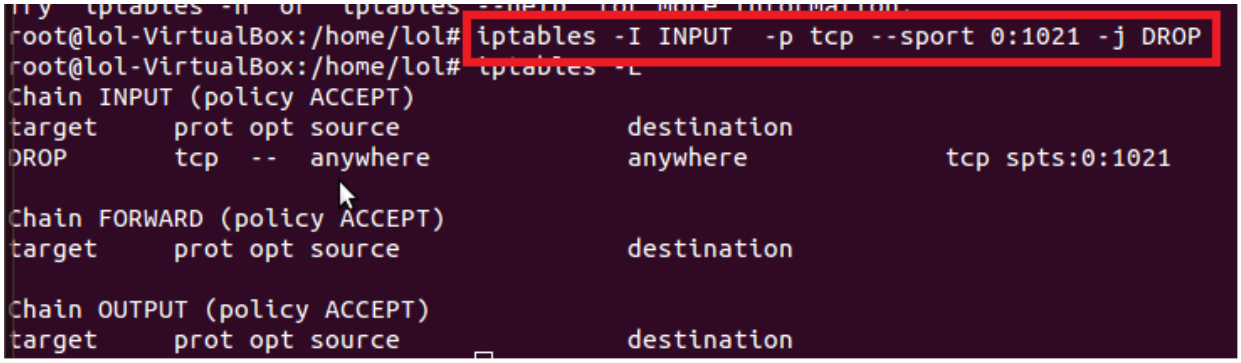


Рис. 2.1

2.2. Дозвіл служби http.

Не пустити віддалених зловмисників у локальну мережу – одна з найважливіших завдань мережевої безпеки, якщо не найважливіша. Цілісність мережі повинна бути захищена від віддалених зловмисників за допомогою точних правил брандмауера. Однак, так як політика за замовчуванням блокує всі вхідні, вихідні та пересилаються пакети, брандмауер/шлюз і користувачі локальної мережі не здатні встановити з'єднання один з одним або з зовнішнім світом. Щоб користувачі виконували пов'язані з мережею функції і

використовували мережеві додатки, адміністратори повинні відкрити певні порти.

Наприклад, щоб дозволити доступ до 80 порту брандмауера потрібно задати правило, що зображене на рисунку нижче. Це дозволить переглядати веб-вміст сайтів, що працюють на порту 80 (http).

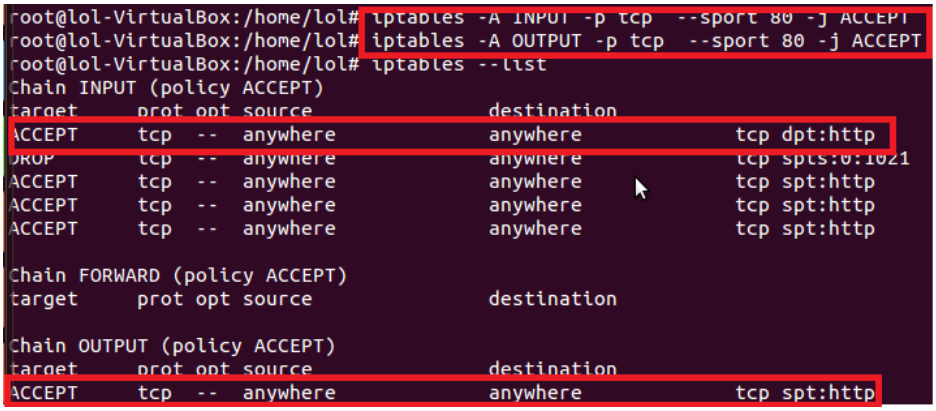


Рис. 2.2

2.3. Дозвіл служби https.

Щоб відкрити доступ до захищених веб-сайтів (наприклад, https://www.example.com/), потрібно відкрити порт 443 (https).

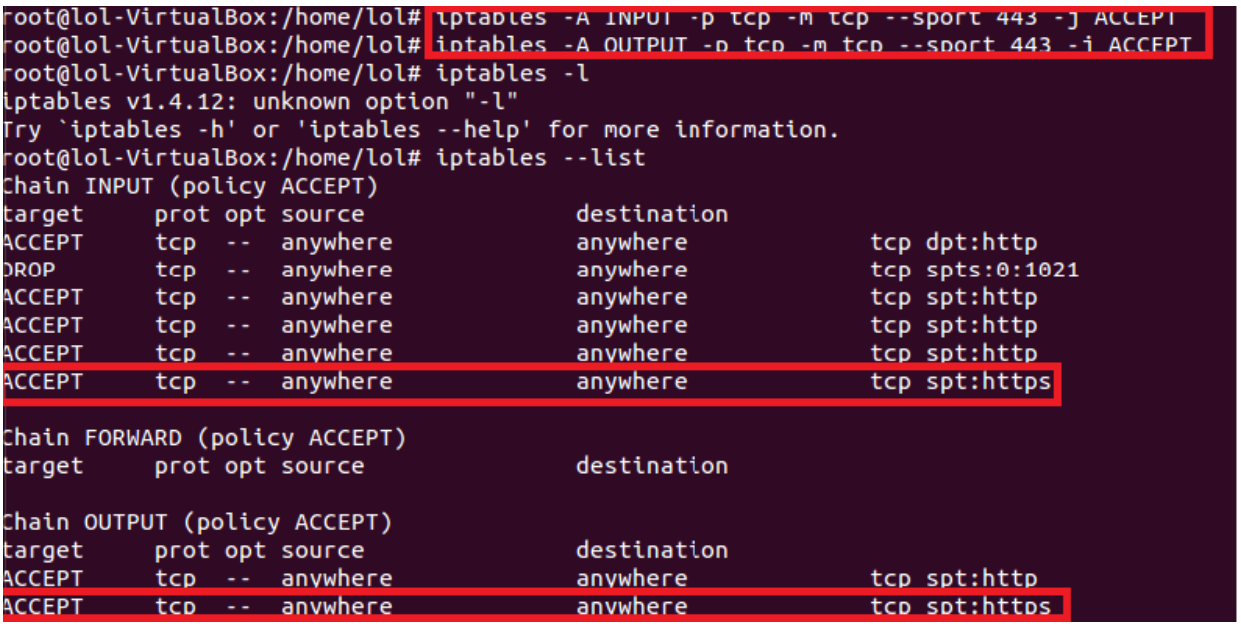


Рис. 2.3

2.4. Дозвіл трафіку IMAP/IMAP2.

В даному правилі ми дозволяємо використання IMAP/IMAP2 трафіку. IMAP надає користувачеві великі можливості для роботи з поштовими скриньками, розташованими на центральному сервері.

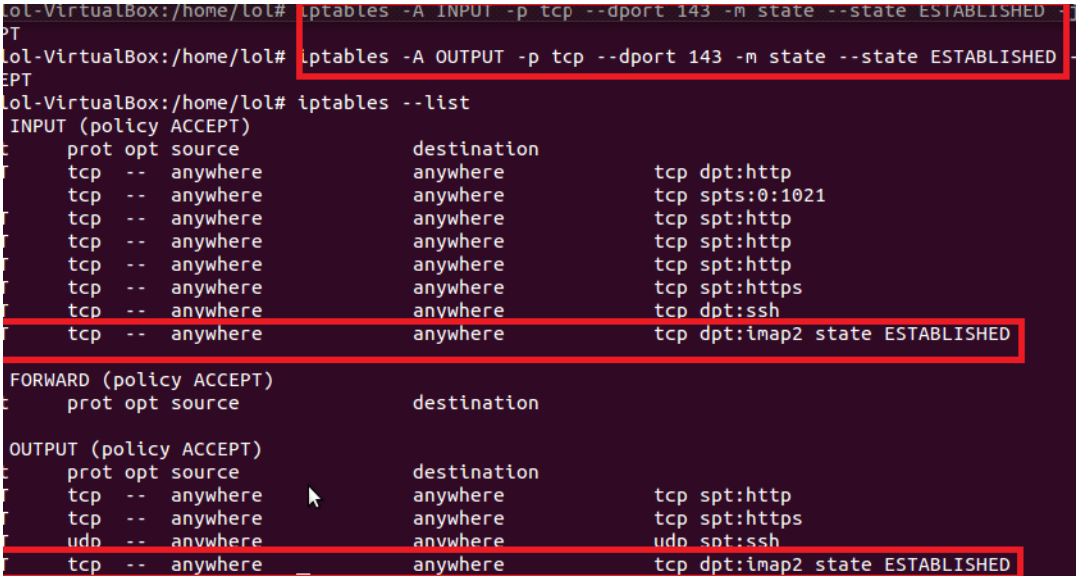


Рис. 2.4

2.5. Використання (-m) state.

У наступному правилі використовується модуль (-m) state, який перевіряє стан встановлюваного сполуки - RELATED або ESTABLISHED і якщо з'єднання підходить під це правило, то дозволяє його.

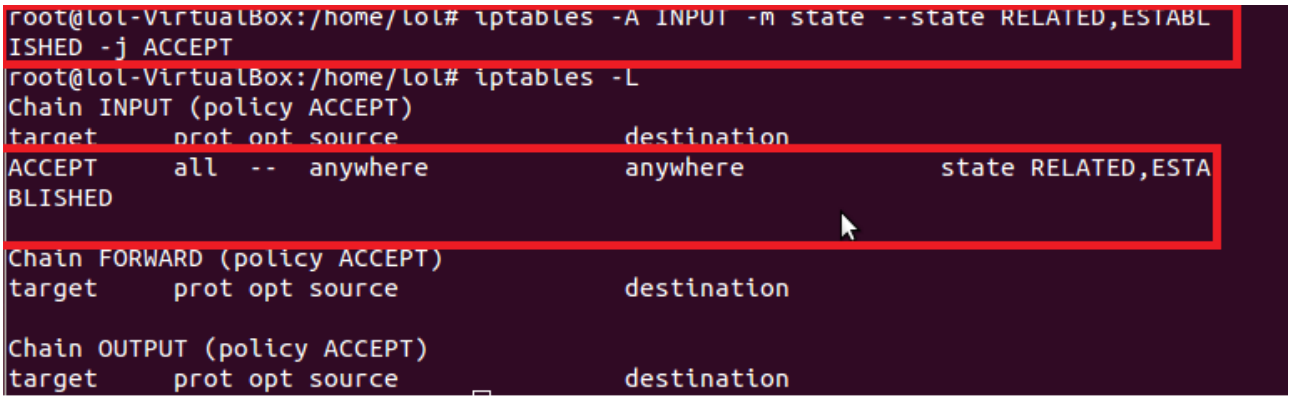


Рис. 2.5

2.6. Правило для SSH

В якості основного правила, ми хочемо дозволити всі встановлені і пов'язані з ними пакети в нашій мережі, і вибірково дозволити через нові пакети в залежності від порту призначення.

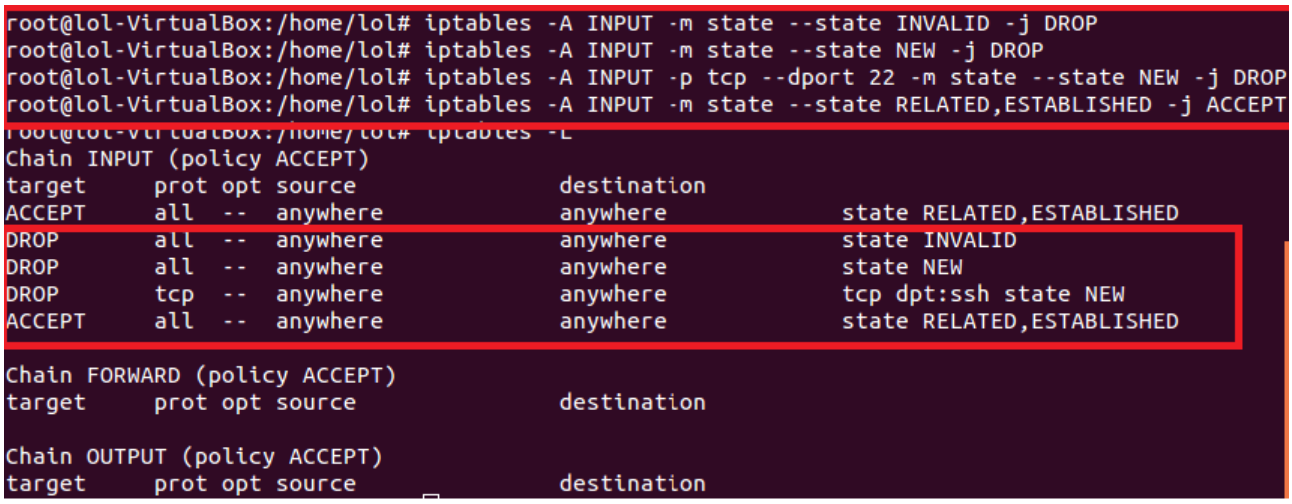
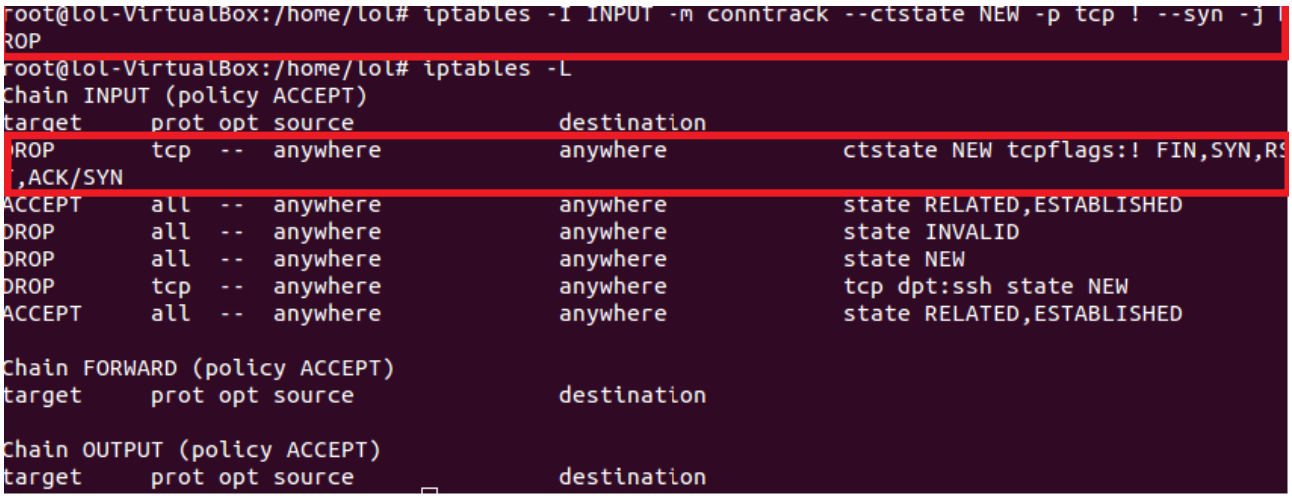


Рис. 2.6

2.7. Правило із SYN-пакетом.

Буде блокувати всі спроби відкрити вхідне TCP-з'єднання не SYN-пакетом. Спроба встановити з'єднання таким чином може бути або помилкою, або атакою.



2.8. Використання –tcp-flags.

В цьому правилі буде створена перешкода спуфінга від нашого імені. Адже якщо ми отримуємо пакет з встановленими прапорами SYN і ACK (такою комбінацією прапорів володіє тільки відповідь на SYN-пакет), то це означає,

що хтось послав іншому хосту SYN-пакет від нашого імені, і відповідь прийшов до нас.

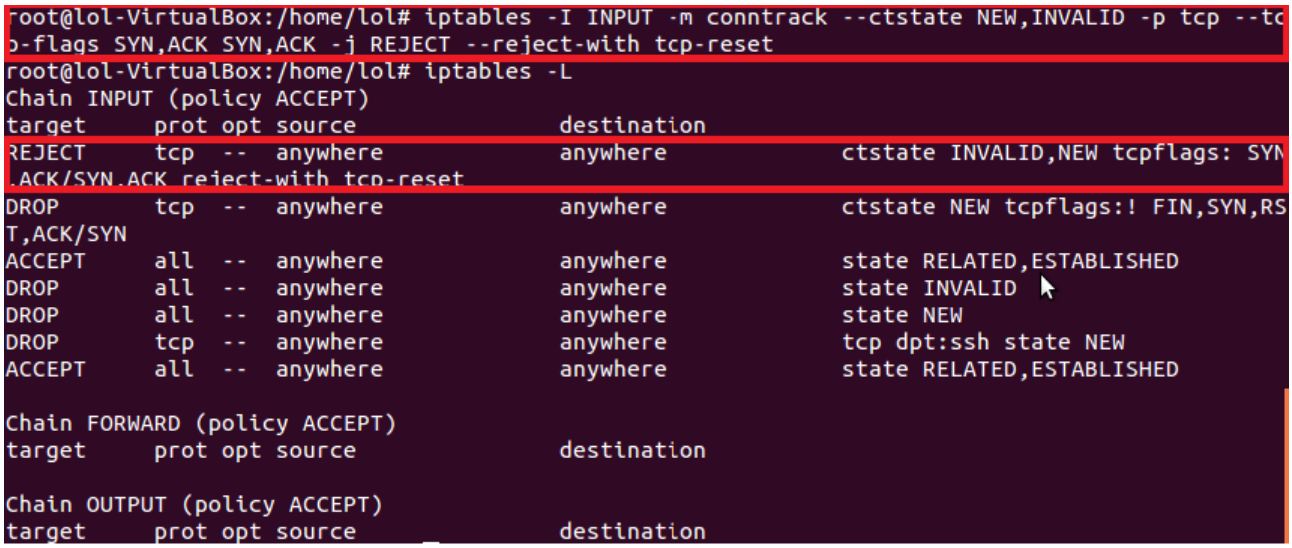


Рис. 2.8

2.9. Дозвіл FTP-серверу.

Дозволено порт 21 (FTP) щодо підключення до міжмережевого екрану.

Рис. 2.9

2.10. Використання limit.

Задає обмеження на кількість пакетів в секундах по певному порту.

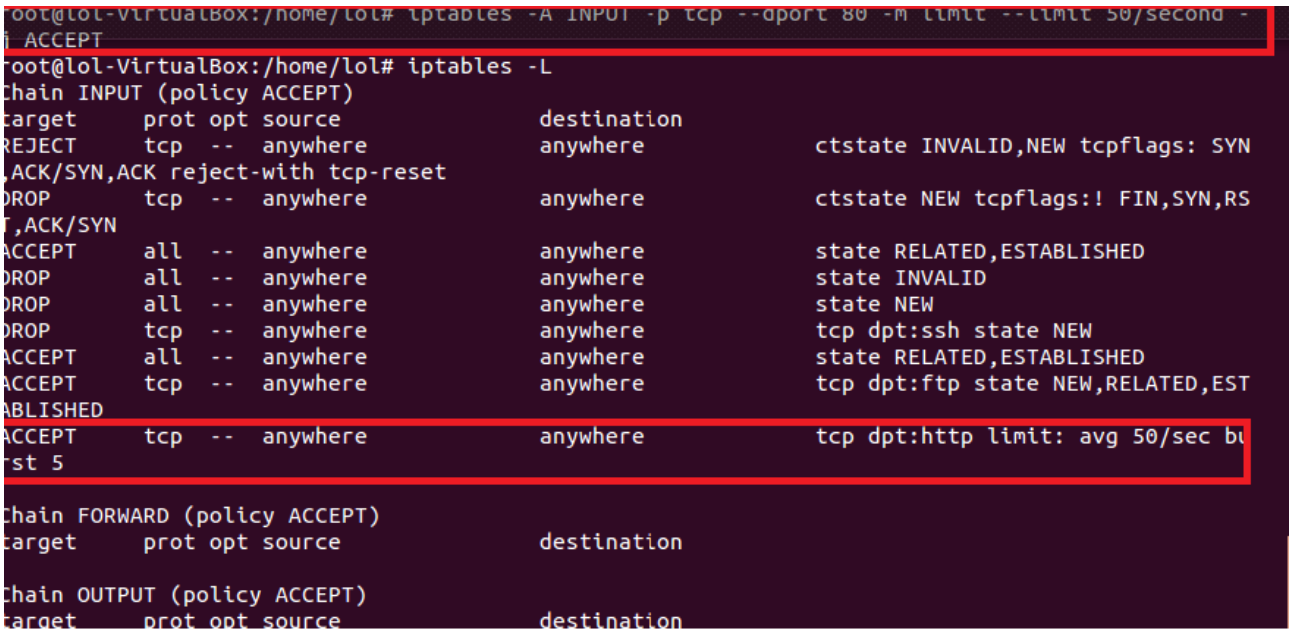


Рис. 2.10

2.11. Робота фільтру.

Щоб фільтр пакетів почав працювати, треба записати '1' в файл / proc/sys/net/ipv4/ip_forward.

Відповідно, якщо туди записати 0, то він працювати перестане. Цей рядок рекомендується вписати в який-небудь скрипт, що автоматично запускається при завантаженні системи.

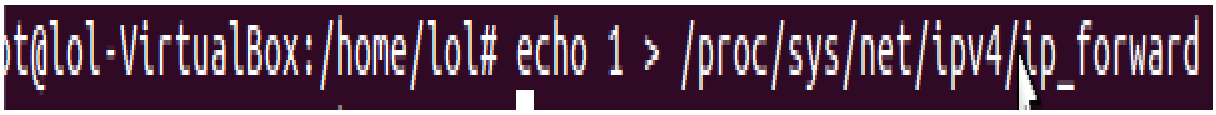


Рис. 2.11

Висновок

Отже, ми детально розглянули тему:”Фільтрація трафіку брандмауером iptables з врахуванням стану TCP-пакетів”. В цій темі було показано, в першу чергу, для чого ця утиліта є створена і як її можна широко застосовувати.

Також було розкрито теоретичну частину, в якій описувалась важливість iptables. Було використано синтаксис iptables у формі таблиці для кращого запам’ятовування. В даному матеріалі також показано суть TCP-пакета з використанням рисунків.

В практичній частині показано, що правила можуть забороняти або дозволяти певну дію. Були наведені приклади, де дозволялося або заборонялося використання трафіка для певних служб (http/https). Утиліта iptables була створена для безпечності комп’ютера. Наприклад, користувач може заборонити приймати пакети, в яких міститься спам.

Електронні ресурси:

1. Дилевский А. Фильтрация пакетов, firewall и маскарадинг в Линуксе [Електронний ресурс]. – Режим доступу: http://citforum.ru/operating_systems/articles/masquerade.shtml#7
2. Захаров И. Протокол TCP №1 [Електронний ресурс]. – Режим доступу: http://www.xakep.ru/post/14943/
3. Лекции по Unix - файл Unix27-Занятие №27. Фильтрация пакетов с помощью IPTABLES (в UNIX) [Електронний ресурс]. – Режим доступу: http://gendocs.ru/v37159/?cc=18
4. Andreasson O. Руководство по iptables (Iptables Tutorial 1.1.19) [Електронний ресурс]. – Режим доступу: http://www.opennet.ru/docs/RUS/iptables/
5. Iptables [Електронний ресурс]. – Режим доступу: http://uk.wikipedia.org/wiki/Iptables
6. Iptables [Електронний ресурс]. – Режим доступу: http://wiki.archlinux.org/index.php/Iptables_%28%D0%A0%D1%83%D1%81%D1%81%D0%BA%D0%B8%D0%B9%29