ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЗАХИСТУ ІНФОРМАЦІЇ

КАФЕДРА ІНФОРМАЦІЙНОЇ ТА КІБЕРНЕТИЧНОЇ БЕЗПЕКИ

Пояснювальна записка

до магістерської роботи
на тему:

«ТЕХНОЛОГІЯ ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ КОРПОРАТИВНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ НА БАЗІ РІШЕННЯ NFTABLES»

Виконав студент 6 курсу, групи БСДМ-62
спеціальності 125 Кібербезпека
освітньо-професійної програми «Інформаційна та кібернетична безпека»
(шифр і назва спеціальності)
Петросян В.Е.
(прізвище та ініціали)
Керівник Борсуковський Ю.В.
(прізвище та ініціали)
Рецензент
(прізвище та ініціали)
Нормоконтролер Чумак Н.С.
(прізвище та ініціали)

КИЇВ – 2023

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

Інститут ННІЗІ
Кафедра Інформаційної та кібернетичної безпеки
Ступінь вищої освіти Магістр
Спеціальність 125 Кібербезпека
Освітньо-професійна програма Інформаційна та кібернетична безпека

ЗАТВЕРДЖУЮ
Завідувач кафедри ІКБ
_____ Гайдур Г.І.
“___”_____2022 року

З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ

Петросяну Вадиму Едуардовичу
(прізвище, ім’я, по батькові)

1. Тема магістерської роботи: «Технологія забезпечення безпеки корпоративної інформаційної системи на базі рішення Nftables»

Керівник магістерської роботи Борсуковський Юрій Володимирович, к.т.н.
(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)
затверджені наказом закладу вищої освіти від «12» жовтня 2022 року №122.

2. Строк подання студентом магістерської роботи 15.12.2022 р.

3. Вихідні дані до магістерської роботи
корпоративна інформаційна система;
програмні комплекси управління захистом інформаційних систем;
наукова та технічна література, експлуатаційна документація, нормативні;
документи, міжнародні стандарти.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1. Актуальність проблеми захисту за допомогою nftables корпоративних інформаційних систем.
2. Склад та умови функціонування корпоративної інформаційної системи.
3. Методи та засоби управління захистом за допомогою nftables.
4. Варіант топології системи управління захистом за допомогою nftables корпоративної інформаційної системи.
5. Перелік графічного матеріалу

1. Тема магістерської роботи.

2.

3.

4.

5.

6.

7.

8.

9.

10.

6. Дата видачі завдання 26.09.2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ зп	Назва етапів магістерської роботи	Строк виконання етапів магістерської роботи	Примітка
1.	Визначення актуальності проблеми захисту кінцевих точок корпоративних інформаційних систем.	26.09.2022 р.	
2.	Аналіз наукової та технічної літератури з питань теми магістерської роботи.	01.10.2022 р.	
3.	Аналіз методів та засобів захисту кінцевих точок.	06.10.2022 р.	
4.	Розроблення варіанту топології системи управління захистом кінцевих точок корпоративної інформаційної системи.	10.10.2022 р.	
5.	Розроблення рекомендацій щодо застосування технології управління захистом кінцевих точок корпоративної інформаційної системи.	11.11.2022 р.	
6.	Оформлення результатів дослідження. Проходження плагіату	14.12.2022 р.	
7.	Підготовка доповіді до захисту.	15.12.2022 р.	

Студент _____ Петросян В.Е.
(підпис) прізвище та ініціали

Керівник магістерської роботи _____ Борсуковський Ю.В.
(підпис) прізвище та ініціали

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

ПОДАННЯ

ГОЛОВІ ДЕРЖАВНОЇ ЕКЗАМЕНАЦІЙНОЇ КОМІСІЇ

ЩОДО ЗАХИСТУ МАГІСТЕРСЬКОЇ РОБОТИ

Направляється студент Петросян В.Е. до захисту магістерської роботи
(прізвище та ініціали)

спеціальності 125 Кібербезпека

освітньо-професійної програми Інформаційна та кібернетична безпека
(шифр і назва спеціальності)

на тему: «Технологія забезпечення безпеки корпоративної інформаційної системи на базі рішення Nftables».

Магістерська робота і рецензія додаються.

Директор інституту _____ _____ Савченко В.А.
(підпис) (прізвище та ініціали)

Довідка про успішність

Петросян В.Е. за період навчання в інституті
(прізвище та ініціали студента)

ННІЗІ з 2021 року по 2023 рік повністю виконав навчальний план за напрямом підготовки, спеціальністю з таким розподілом оцінок за:
національною шкалою: відмінно ____%, добре ____%, задовільно ____%;
шкалою ECTS: A ____%; B ____%; C ____%; D ____%; E ____%.

Секретар інституту _____ Бреславська Т.В.
(підпис) (прізвище та ініціали)

Висновок керівника магістерськоїї роботи

Студент ***Петросян В.Е.*** обрав тему роботи, метою якої було дослідити зміст технології забезпечення безпеки корпоративної інформаційної системи на базі рішення Nftables та розробити варіант топології системи управління їх захистом на підприємстві. Перелік використаних джерел свідчить про вміння магістром розбиратись в наукових питаннях та застосовувати їх при дослідженнях. Під час виконання магістерської роботи Петросян В.Е. показав відмінну теоретичну та практичну підготовку, вміння самостійно вирішувати питання і робити висновки. Роботу виконував сумлінно, акуратно та вчасно за планом.

Все це дозволяє оцінити виконану магістерську роботу студента Петросяна Вадима Едуардовича на оцінку **«добре»** та присвоїти йому кваліфікацію магістр з кібербезпеки за спеціалізацією інформаційна та кібернетична безпека.

Керівник магістерської роботи _____ Борсуковський Ю.В.
(підпис) (прізвище та ініціали)

“____” _____ 2022 року

Висновок кафедри про магістерську роботу

Магістерська робота розглянута. Студент Петросян В.Е.
(прізвище та ініціали)

допускається до захисту даної магістерської роботи в Державній екзаменаційній комісії

Завідувач кафедри Інформаційної та кібернетичної безпеки
(назва)

_____ Гайдур Г.І.
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Текстова частина магістерської роботи: 64 сторінки, 35 рисунків, 1 таблиці, 14 джерел.

Об'єкт дослідження – процес забезпечення безпеки корпоративної інформаційної мережі на базі рішення Nftables.

Предмет дослідження – технологія управління безпеки корпоративної інформаційної мережі на базі рішення Nftables.

Мета роботи – розробити варіант топології системи управління захистом корпоративної інформаційної мережі за допомогою Nftables та рекомендації щодо застосування технології їх захисту на підприємстві.

Методи дослідження – опрацювання літератури за даною темою, аналіз міжнародних стандартів, експлуатаційної документації та їх порівняння, моделювання процесу управління захистом корпоративної інформаційної мережі за допомогою брандмауера.

В роботі проведено аналіз проблеми забезпечення кібербезпеки корпоративної інформаційної системи та визначено завдання та мета управління захистом корпоративної інформаційної мережі. Проаналізовано існуючі технології захисту корпоративної інформаційної мережі.

Досліджено методи та засоби управління захистом корпоративної мережі на прикладі рішення Nftables. Визначено основні функції та склад програмного комплексу Nftables. Визначено призначення, основні функції та принципи роботи рішень на основі Nftables.

На основі досліджень проведених в роботі розроблено варіант топології системи управління захистом корпоративної інформаційної системи та рекомендації щодо застосування технології управління їх захистом на підприємстві.

Галузь використання – кібербезпека корпоративної інформаційної мережі.

КОРПОРАТИВНА ІНФОРМАЦІЙНА МЕРЕЖА, КІБЕРБЕЗПЕКА, ЗАХИСТ МЕРЕЖІ, ЗАСОБИ ТА МЕТОДИ УПРАВЛІННЯ ЗАХИСТОМ КОРПОРАТИВНОЇ МЕРЕЖІ, ТЕХНОЛОГІЯ УПРАВЛІННЯ ЗАХИСТОМ МЕРЕЖІ

ABSTRACT

Master's thesis: 64 pages, 35 figures, 1 tables, 14 sources.

Object of research – the process of ensuring the security of the corporate information system based on the Nftables solution.

Subject of research – the security management technology of the corporate information system based on the Nftables solution.

The aim of research – to develop a version of the topology of the protection management system of the corporate information system and recommendations for the application of technology for their protection at the enterprise.

Research methods – elaboration of literature on the topic, analysis of operational documentation, international standards and their comparison, modeling of the process of managing the protection of the endpoints of the corporate information system.

The paper analyzes the problem of ensuring cyber security of the corporate information system and defines the task and purpose of managing the protection of the corporate information network. The existing technologies for protecting the corporate information network are analyzed.

The methods and means of managing the protection of the corporate network were studied using the Nftables solution as an example. The main functions and composition of the Nftables software complex are defined. The purpose, main functions and operating principles of solutions based on Nftables are defined.

On the basis of the research carried out in the work, a version of the topology of the protection management system of the corporate information system and recommendations for the application of technology for managing their protection at the enterprise were developed.

Field of use – cyber security of the corporate information network.

CORPORATE INFORMATION NETWORK, CYBER SECURITY, NETWORK PROTECTION, TOOLS AND METHODS OF CORPORATE NETWORK PROTECTION MANAGEMENT, NETWORK PROTECTION MANAGEMENT TECHNOLOGY

ЗМІСТ

ВСТУП 9

1 АНАЛІЗ ОСНОВНИХ МЕХАНІЗМІВ ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ КОРПОРАТИВНИХ МЕРЕЖ ВІД ЗАГРОЗ 11

1.1 Загрози інформацій безпеці корпоративної мережі підприємства 11

1.2 Історія походження міжмережевого екрану 21

1.3 Види міжмережевого екрану 22

1.4 Використання міжмережевого екрану на прикладі апаратного брандмауеру 23

1.5 Як працює міжмережевий екран – Netfilter в Linux 28

1.6 Файрвол або фільтрація пакетів 30

2 АНАЛІЗ РОБОЧОГО ПРОСТОРУ ТА СТВОРЕННЯ ВІРТУАЛЬНОГО СЕРЕДОВИЩА 33

2.1 Аналіз робочого простору віртуальної мережі 33

2.2 Створення віртуальних середовищ 34

3 ЗАСТОСУВАННЯ ТЕХНОЛОГІЇ БЕЗПЕКИ КОРПОРАТИВНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ 41

3.1 Встановлення лабораторного стенду 41

3.2 Захист робочої станції за допомогою локального брандмауера та розгляд моделі загроз 43

3.3 Схеми розмежування трафіку за рівнем захищеності та простоти реалізації 47

3.4 Створення лабораторного стенду корпоративної інформаційної системи 49

3.5 Використання Nftables правил 52

ВИСНОВКИ 60

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ 62

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (ПРЕЗЕНТАЦІЯ) 64

ВСТУП

Проблема інформаційної безпеки у корпоративних мережах зв'язку сьогодні дуже гостро постає перед компаніями будь-якого рівня. Витік критично важливої корпоративної інформації, зростання обсягів паразитного трафіку, здирство, шантаж та замовні атаки на інформаційні ресурси стали останнім часом частим явищем.

Важливість інтернету для будь-якого сучасного підприємства – зрозуміла та незаперечна. Інтернет служить для комунікації з партнерами та замовниками (у тому числі через корпоративний веб-сайт), забезпечення віддаленого доступу до корпоративних ресурсів та гнучкого розширення робочого простору, пошуку корисної інформації та здійснення електронних комерційних транзакцій. Тим часом інтернет має ряд властивостей, які ускладнюють забезпечення інформаційної безпеки:

інтернет - це публічна відкрита мережа з нецентралізованою топологією та маршрутизацією;

шкідлива активність може виникнути в одній частині інтернету, а потім швидко поширитися по всій Всесвітній мережі;

в інтернеті контролюється головним чином вхідний трафік, але не вихідний;

у Світовій мережі практично відсутня ідентифікація користувачів;

юрисдикція країни, у якій стався злочин, часто не поширюється на кіберзлочинця.

Інтернет - це середовище, в якому інформаційне безпека підприємств наражається на найбільші загрози, і водночас це важливий засіб здійснення бізнес-процесів.

Найактуальніші загрози інформаційної безпеки в інтернеті - це:

спрямовані на певне підприємство або галузь (таргетовані) хакерські та вірусні атаки;

крадіжки корпоративних даних внаслідок атак на мобільні пристрої;

зараження шкідливими програмами та розголошення конфіденційної інформації у соціальних мережах;

непомітне інфікування комп'ютерів та інших пристроїв під час відвідування безпечних на перший погляд веб-сайтів (атаки Drive-by);

використання недостатньо захищених хмарних веб-сервісів та технології SaaS (Software as a Service, "ПЗ як сервіс").

1 АНАЛІЗ ОСНОВНИХ МЕХАНІЗМІВ ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ КОРПОРАТИВНИХ МЕРЕЖ ВІД ЗАГРОЗ

1.1 Загрози інформацій безпеці корпоративної мережі підприємства

На даний момент існує безліч підсистем інформаційної безпеки, тому в даному рефераті зупинимося тільки на деяких конкретних видах, що належать до ринку корпоративних засобів мережевої безпеки, виключаючи засоби захисту для домашніх мереж. Фахівці стосовно даного ринку відзначають такі види технологій та пристроїв захисту:

- Міжмережеві екрани UTM/Firewall;
- Системи запобігання вторгненням IPS/IDS;
- Системи захисту від розподілених атак DDoS;
- Контроль доступу до мережі NAC;
- Віртуальні приватні мережі VPN;
- Аутентифікації та авторизації користувачів AAA;
- Системи керування доступом IDM
- Міжмережеві екрани

Міжмережевий екран (мережевий екран) - це комплекс апаратних та програмних засобів комп'ютерної мережі, що здійснює контроль та фільтрацію проходять через нього мережевих пакетів відповідно до заданих правил.

Основним завданням мережевого екрана є захист мережі або окремих її вузлів від несанкціонованого доступу. Також мережні екрани часто називають фільтрами, тому що їх основне завдання – не пропускати (фільтрувати) пакети, які не підходять під критерії, визначені у конфігурації.

Деякі мережеві екрани також дозволяють здійснювати трансляцію адрес динамічну заміну внутрішньо-мережевих (сірих) адрес або портів на зовнішні, які використовуються за межами локальної мережі, - що може забезпечувати додаткову безпеку.

Системи запобігання вторгненням IPS/IDS

Intrusion Detection System (IDS) або Intrusion Prevention System (IPS) - це програмні та апаратні засоби для виявлення та запобігання вторгненням. Вони призначені для виявлення та запобігання спробам несанкціонованого доступу, використання або виведення з ладу комп'ютерних систем, головним чином через Інтернет або локальну мережу. Такі спроби можуть мати форму як атаки хакерів чи інсайдерів, і бути результатом дій шкідливих програм.

В цілому IPS за класифікацією та своїми функціями аналогічні IDS. Головна відмінність IPS від IDS полягає в тому, що IPS функціонують у реальному часі і можуть автоматично блокувати мережеві атаки. Кожна IPS включає модуль IDS.

Системи захисту від розподілених атак DDoS;

Фахівці розрізняють кілька способів та видів групування DoS-атак за типами. Основними двома групами DoS-атак є руйнівні та атаки на ресурси системи.

Руйнівники - це атаки, які призводять до того, що пристрій у мережі стає повністю непрацездатним: зависає, знищується операційна система чи конфігурація. Такі атаки засновані на вразливості ПЗ систем, що атакуються.

Атаки на ресурси системи – це атаки, які значно знижують продуктивність пристроїв чи додатків.

Дуже поширеною є атака з метою заповнення пропускної спроможності каналів зв'язку.

Останнім часом набули поширення DDOS-атаки за допомогою створення мережі шкідливих ботів (ботнетів). Бот - спеціальна програма, що виконує автоматично та/або за заданим розкладом будь-які дії через ті ж інтерфейси, що і звичайний користувач.

Основні шкідливі дії роботів:

- Спам-боти, що збирають адреси E-mail із контактних форм та гостьових книг;

- Програми, що завантажують інтернет канал потоком непотрібної інформації (як правило, рекламного характеру);
- Сайти, що збирають інформацію про нешкідливі сайти, для використання її в автоматично створюваних;
- Деякі віруси та черв'яки;
- DoS- та DDoS-атаки;
- Ботнети та комп'ютери-зомбі.

Система захисту від DDoS-атак базується на маршрутизаторах, що вже є в мережі, і додає в мережу свої два компоненти:

- Пристрій для блокування DDoS-атаки. В англійській мові цей пристрій називають mitigator – блокатор;
- Пристрій із вбудованим штучним інтелектом для виявлення DDoS-атаки та перенаправлення атаки на блокатор – детектор.

При цьому завдання блокатора входить не тільки блокування трафіку, але і його уповільнення. Після виявлення DDoS-атаки на якусь мережу аналізатор трафіку вставляє в таблиці динамічної маршрутизації (за допомогою BGP або OSPF) запис, який говорить, що маршрут в мережу, що атакується, лежить через цей блокатор.

В результаті весь трафік атаки починає проходити через блокатор, що дає можливість заблокувати трафік атаки, а легітимний трафік передати в мережу, що захищається. Передача в мережу здійснюється будь-яким доступним способом, наприклад, за допомогою інкапсуляції трафіку всередині GRE.

Після завершення атаки таблиця маршрутизації перелаштовується, щоб трафік проходив через кінцевий маршрутизатор, пов'язаний з цією мережею.

Контроль доступу до мережі NAC

В останні роки все більш помітне місце на ринку систем безпеки займає ефективна стратегія захисту за допомогою контролю доступу до мережі та її ресурсів - технологія NAC (Network Access Control).

Суть цієї стратегії захисту зводиться до регламенту доступу користувачів до мережі та постійного контролю за станом кінцевих мережевих пристроїв - як внутрішніх співробітників, так і віддалених, у тому числі мобільних користувачів.

Технологія NAC реалізує широкий перелік функцій, що охоплюють ідентифікацію користувачів, оцінку та ідентифікацію стану кінцевих точок, запобіжні заходи, перевірку пристроїв на відповідність корпоративним політикам доступу в мережу.

На ранньому етапі розвитку NAC методи контролю підключень персональних пристроїв до мережі реалізувалися на основі протоколу 802.1x і були надзвичайно складні для впровадження та підтримки в масштабних мережах.

Застосування системи контролю доступу на основі протоколу 802.1x дозволяє:

- контролювати підключення користувачів чи пристроїв до мережі (ідентифікувати);
- призначити в певний VLAN, присвоїти відповідну IP-адресу;
- збирати статистику: який користувач, коли, на якому комутаторі, з якою IP-адресою здійснив підключення до мережі. Опціонально - скільки часу пропрацював, яке навантаження на мережу зробив;
- надати гостьовий вхід до Інтернету для робочих станцій замовників та партнерів, що знаходяться на території організації;
- забезпечити підключення принтерів, терміналів та інших пристроїв, що не підтримують 802.1x.

Таким чином, застосування 802.1x гарантує, що в мережі ніколи не з'явиться неавторизований пристрій. Однак не слід забувати, що при 802.1x контроль доступу до знімних носіїв і перевірка програмного забезпечення пристроїв не підтримуються. Адже саме ці два фактори є причиною виникнення більшості інцидентів (відплив інформації, поширення вірусів у локальній мережі та ін.). З розвитком другого покоління систем контролю доступу до мережі прийшли і рішення, що ґрунтуються на використанні агента, який контролює

цілісність програмного середовища та процесів, що протікають на робочих станціях. Вже на цьому рівні виникає питання профілювання різних користувачів та застосування до них різних політик доступу.

Віртуальні приватні мережі VPN

VPN (англ. Virtual Private Network – віртуальна приватна мережа) – логічна мережа, створювана поверх іншої мережі, наприклад Інтернет. Незважаючи на те, що комунікації здійснюються по публічних мережах з використанням небезпечних протоколів, за рахунок шифрування створюються закриті від сторонніх канали обміну інформацією. VPN дозволяє об'єднати, наприклад, кілька офісів організації в єдину мережу з використанням зв'язку між ними непідконтрольних каналів.

За допомогою методики тунелювання пакети даних транслюються через загальнодоступну мережу як за звичайним двоточковим з'єднанням. Між кожною парою "відправник-одержувач даних" встановлюється своєрідний тунель - безпечне логічне з'єднання, що дозволяє інкапсулювати дані одного протоколу пакети іншого.

Найбільш поширений метод створення тунелів VPN - інкапсуляція мережевих протоколів (IP, IPX, AppleTalk і ін.) у PPP та подальша інкапсуляція утворених пакетів у протокол тунелювання. Зазвичай як останні виступає IP або (набагато рідше) ATM і Frame Relay. Такий підхід називається тунелюванням другого рівня, оскільки "пасажиром" тут є протокол саме другого рівня.

Альтернативний підхід - інкапсуляція пакетів мережного протоколу безпосередньо в протокол тунелювання (наприклад, VTP) називається тунелюванням третього рівня.

Як правило, на сьогоднішній день для побудови мереж VPN використовуються протоколи наступних рівнів:

- Канальний рівень
- Мережевий рівень
- Транспортний рівень

Існують такі технології аутентифікації:

1) Мережева аутентифікація на основі багаторазового пароля
2) Аутентифікація із використанням одноразового пароля
3) Аутентифікація з урахуванням сертифікатів
4) Аутентифікація мережі на основі багаторазового пароля.

Відповідно до базового принципу "єдиного входу", коли користувачеві достатньо один раз пройти процедуру автентифікації, щоб отримати доступ до всіх мережевих ресурсів, у сучасних операційних системах передбачаються централізовані служби автентифікації. Така служба підтримується одним із серверів мережі та використовує для своєї роботи базу даних, в якій зберігаються облікові дані (іноді звані бюджетами) про користувачів мережі. Облікові дані містять поряд з іншою інформацією ідентифікатори та паролі користувачів. Спрощено схема аутентифікації у мережі виглядає так. Коли користувач здійснює логічний вхід до мережі, він набирає на клавіатурі свого комп'ютера свої ідентифікатор та пароль. Ці дані використовуються службою аутентифікації - в централізованій базі даних, що зберігається на сервері, за ідентифікатором користувача знаходиться відповідний запис, з нього вилучається пароль та порівнюється з тим, який ввів користувач. Якщо вони збігаються, то автентифікація вважається успішною, користувач отримує легальний статус і права, які визначені для нього системою авторизації.

Аутентифікація за допомогою одноразового пароля.

Алгоритми аутентифікації, що базуються на багаторазових паролях, не дуже надійні. Паролі можна піддивитися або просто вкрасти. Більш надійними виявляються схеми, які використовують одноразові паролі. З іншого боку, одноразові паролі набагато дешевші і простіші за біометричні системи аутентифікації, такі як сканери сітківки ока або відбитків пальців. Все це робить системи, що базуються на одноразових паролях, дуже перспективними. Слід пам'ятати, що, зазвичай, системи аутентифікації з урахуванням одноразових паролів розраховані на перевірку лише віддалених, а чи не локальних користувачів. Генерація одноразових паролів може виконуватись або програмно, або апаратно. Деякі реалізації апаратних пристроїв доступу на основі

одноразових паролів є мініатюрними пристроями з вбудованим мікропроцесором, схожі на звичайні пластикові картки, які використовуються для доступу до банкоматів. Такі картки, які часто називають апаратними ключами, можуть мати клавіатуру і маленьке дисплейне вікно. Апаратні ключі можуть бути також реалізовані у вигляді пристрою, що приєднується до роз'єму, що розташовується між комп'ютером і модемом, або у вигляді карти (гнучкого диска), що вставляється в дисковод комп'ютера. Існують і програмні реалізації засобів аутентифікації на основі одноразових паролів (програмні ключі). Програмні ключі розміщуються на змінному магнітному диску як звичайної програми, важливою частиною якої є генератор одноразових паролів. Застосування програмних ключів і карток, що приєднуються до комп'ютера, пов'язане з деяким ризиком, оскільки користувачі часто забувають гнучкі диски в машині або не від'єднують картки від ноутбуків. Незалежно від того, яку реалізацію системи аутентифікації на основі одноразових паролів вибирає користувач, він, як і в системах аутентифікації з використанням багаторазових паролів, повідомляє системі свій ідентифікатор, проте замість того, щоб вводити щоразу той самий пароль, він вказує послідовність цифр , що повідомляється йому апаратним або програмним ключем. Через певний період часу генерується інша послідовність - новий пароль. Аутентифікаційний сервер перевіряє введену послідовність та дозволяє користувачеві здійснити логічний вхід. Аутентифікаційний сервер може являти собою окремий пристрій, виділений комп'ютер або програму, що виконується на звичайному сервері. як і в системах автентифікації з використанням багаторазових паролів, повідомляє системі свій ідентифікатор, проте замість того, щоб вводити щоразу один і той же пароль, він вказує послідовність цифр, що повідомляється йому апаратним або програмним ключем. Через певний період часу генерується інша послідовність - новий пароль. Аутентифікаційний сервер перевіряє введену послідовність та дозволяє користувачеві здійснити логічний вхід. Аутентифікаційний сервер може являти собою окремий пристрій, виділений комп'ютер або програму, що виконується на звичайному сервері. як і в системах автентифікації з використанням

багаторазових паролів, повідомляє системі свій ідентифікатор, проте замість того, щоб вводити щоразу один і той же пароль, він вказує послідовність цифр, що повідомляється йому апаратним або програмним ключем. Через певний період часу генерується інша послідовність - новий пароль. Аутентифікаційний сервер перевіряє введену послідовність та дозволяє користувачеві здійснити логічний вхід. Аутентифікаційний сервер може являти собою окремий пристрій, виділений комп'ютер або програму, що виконується на звичайному сервері. повідомляється йому апаратним чи програмним ключем. Через певний період часу генерується інша послідовність - новий пароль. Аутентифікаційний сервер перевіряє введену послідовність та дозволяє користувачеві здійснити логічний вхід. Аутентифікаційний сервер може являти собою окремий пристрій, виділений комп'ютер або програму, що виконується на звичайному сервері. повідомляється йому апаратним чи програмним ключем. Через певний період часу генерується інша послідовність - новий пароль. Аутентифікаційний сервер перевіряє введену послідовність та дозволяє користувачеві здійснити логічний вхід. Аутентифікаційний сервер може являти собою окремий пристрій, виділений комп'ютер або програму, що виконується на звичайному сервері.

Аутентифікація на основі сертифікатів.

Аутентифікація із застосуванням цифрових сертифікатів є альтернативою використанню паролів і є природним рішенням в умовах, коли кількість користувачів мережі (нехай і потенційних) вимірюється мільйонами. У таких обставинах процедура попередньої реєстрації користувачів, пов'язана з призначенням та зберіганням їх паролів, стає вкрай обтяжливою, небезпечною, а іноді й просто нереалізованою. При використанні сертифікатів мережа, яка дає користувачеві доступ до своїх ресурсів, не зберігає жодної інформації про своїх користувачів - вони її надають у своїх запитах у вигляді сертифікатів, що засвідчують особу користувачів. Сертифікати видаються спеціальними уповноваженими організаціями – центрами сертифікації (Certificate Authority, CA). Тому завдання зберігання секретної інформації (закритих ключів) покладається на самих користувачів, що робить це рішення набагато

масштабованішим, ніж варіант із використанням централізованої бази паролів. Сертифікат є засобом автентифікації Користувача при його зверненні до мережевих ресурсів, роль аутентифікуючої сторони відіграють інформаційні сервери корпоративної мережі або Інтернету. У той же час і сама процедура отримання сертифікату включає етап аутентифікації, аутентифікатором тут виступає сертифікуюча організація. Для отримання сертифіката клієнт повинен повідомити сертифікуючу організацію свій відкритий ключ і ті чи інші відомості, що засвідчують його особу. Всі ці дані клієнт може надіслати електронною поштою або принести на гнучкому диску особисто. Перелік необхідних даних залежить від типу сертифіката. Сертифікуюча організація перевіряє докази автентичності, поміщає свій цифровий підпис у файл, що містить відкритий ключ, і посилає сертифікат назад, підтверджуючи факт належності даного конкретного ключа особі. Після цього сертифікат може бути вбудований у будь-який запит використання інформаційних ресурсів мережі.

У мережевій термінології широко застосовується термін AAA (від англ. Authentication, Authorization, Accounting), який використовується для опису процесу надання доступу та контролю за ним. Як сервер аутентифікації AAA дозволяє використовувати RADIUS чи TACAS+ сервер.

RADIUS (Remote Authenticationin Dial-In User Service) - протокол для реалізації аутентифікації, авторизації та збору відомостей про використані ресурси, розроблений для передачі відомостей між центральною платформою та обладнанням. Цей протокол застосовувався для системи тарифікації використаних ресурсів конкретним користувачем/абонентом.

TACACS+ (Terminal Access Controller Access Control System plus) - сеансовий протокол, результат подальшого вдосконалення TACACS, здійсненого Cisco. Поліпшено безпеку протоколу (шифрування), а також введено поділ функцій автентифікації, авторизації та обліку, які тепер можна використовувати окремо.

Системи керування доступом IDM

Системи управління правами доступу та обліковими записами користувачів класу IDM (Identity Management) або як їх ще називають IAM (Identity and Access Management) призначені для централізованого управління правами на доступ до інформації, обліковими записами, паролями та іншими атрибутами у різних інформаційних системах, що дозволяє автоматизувати процеси управління правами доступу, знизити ризики інформаційної безпеки та оптимізувати витрати на адміністрування IT інфраструктури.

IDM система суттєво полегшує доступ до мережевих інформаційних ресурсів для співробітників організації, при цьому підвищується рівень захисту корпоративних інформаційних систем. Централізована система керування обліковими записами користувачів дозволяє відстежувати всі облікові записи в підключених інформаційних системах.

IDM надає консолідовану звітність про всі облікові записи та права доступу користувачів компанії в інтегрованих з IDM інформаційних системах.

Основні завдання, які вирішує IDM:

- Автоматизація процесу керування ідентифікаційними даними;
- Уніфікація облікових даних користувачів;
- Централізоване управління правами доступу;
- Скорочення витрат на адміністрування інформаційних систем та обслуговування користувачів;
- зниження ризиків несанкціонованого доступу до корпоративних інформаційних систем;
- Скорочення тимчасових витрат на надання, зміну та відкликання прав доступу користувачів;

1.2 Історія походження міжмережевого екрану

Міжмережеве екранування - блокування трафіку від несанкціонованих джерел - одна з найстаріших мережевих технологій безпеки, але виробники відповідних середовищ продовжують розробляти нові підходи, які допомагають ефективніше протидіяти сучасним загрозам в мережевому оточенні, що змінюється, і захищати корпоративні IT-ресурси. Міжмережеві екрани нового покоління дозволяють створювати політики, використовуючи ширший спектр контекстних даних, та забезпечувати їх дотримання.

Історія походження

Еволюція міжмережевих екранів (FW – Firewall) пройшла довгий шлях. Вперше вони були розроблені ще наприкінці 80-х компанією DEC і функціонували переважно на перших чотирьох рівнях моделі OSI, перехоплюючи трафік та аналізуючи пакети на відповідність заданим правилам. Потім Check Point запропонувала міжмережевий екрани зі спеціалізованими мікросхемами (ASIC) для глибокого аналізу заголовків пакетів. Ці вдосконалені системи могли вести таблицю активних з'єднань та задіяли її у правилах (Stateful Packet Inspection, SPI). Технологія SPI дозволяє перевіряти IP-адреси відправника та одержувача та контролювати порти.

Великим кроком уперед вважається створення FW, що функціонують лише на рівні додатків. Перший такий продукт випустила компанія SEAL ще 1991-го, а через два роки з'явилося відкрите рішення Firewall Toolkit (FWTK) від Trusted Information Systems. Ці міжмережеві екрани перевіряли пакети на всіх семи рівнях, що дозволило використовувати в наборах правил (політиках) розширену інформацію - не тільки про з'єднання та їх стан, а й про операції з використанням протоколу конкретної програми. До середини 90-х міжмережевих екранів набули здатність здійснювати моніторинг популярних протоколів прикладного рівня: FTP, Gopher, SMTP і Telnet. Ці удосконалені продукти (application-aware) одержали назву міжмережевих екранів нового покоління (Next-Generation Firewall, NGFW).

TIS випустила комерційну версію FWTK – Gauntlet Firewall. Саме цей продукт, що володіє можливостями автентифікації користувачів, фільтрації URL, а також засобами захисту від шкідливих програм та функціями безпеки рівня програм, вважається першим міжмережевим екраном нового покоління. Таким чином, формально продуктам NGFW вже понад 15 років, хоча сьогодні в цей термін вкладають інший зміст.

1.3 Види міжмережевого екрану

Підтримуваний рівень мережевої моделі OSI є основною характеристикою класифікації міжмережевих екранів. Розрізняють такі типи міжмережевих екранів:

1) Керовані комутатори(канальний рівень);
2) Фільтри міжмережевого рівня (stateless). Статична фільтрація здійснюється шляхом аналізу IP-Адреси джерела та приймача, протоколу, портів відправника та одержувача;
3) Шлюзи сеансового рівня (circuit-level proxy). У мережній моделі TCP/IP немає рівня, що однозначно відповідає сеансовому рівню OSI, тому до шлюзів сеансового рівня відносять фільтри, які неможливо ототожнити ні з мережевим, ні з транспортним, ні з прикладним рівнем:
   - Шлюзи, що транслюють адреси (NAT,PAT) або мережеві протоколи (транслюючий міст);
   - Фільтри контролю стану каналу. До фільтрів контролю стану каналу зв'язку нерідко відносять мережеві фільтри мережевого рівня з розширеними можливостями (stateful), які додатково аналізують заголовки пакетів і вміють фільтрувати фрагментовані пакети);
   - Шлюзи сеансового рівня. Найбільш відомим та популярним шлюзом сеансового рівня є посередник SOCKS;

4) Шлюзи прикладного рівня (application-level proxy), часто звані проксі-серверами. Поділяються на прозорі (transparent) та непрозорі (solid).
5) Брандмауер SPI (Stateful Packet Inspection, SPI), або інакше брандмауери з динамічною фільтрацією пакетів (Dynamic Packet Filtering) є по суті шлюзами сеансового рівня з розширеними можливостями. Інспектори стану оперують на сеансовому рівні, але "розуміють" протоколи прикладного та мережевого рівнів. На відміну від шлюзу прикладного рівня, що відкриває два віртуальні канали TCP (один - для клієнта, інший - для сервера) для кожного з'єднання, інспектор стану не перешкоджає організації прямого з'єднання між клієнтом та сервером.

Існує також поняття "міжмережевий екран експертного рівня". Мережевий екран базується на посередниках прикладного рівня або інспекторах стану, але обов'язково комплектується шлюзами сеансового рівня та мережевими фільтрами, іноді розуміючи і мережевий рівень. Найчастіше мають систему протоколювання подій та оповіщення адміністраторів, засоби підтримки віддалених користувачів (наприклад, авторизація), засоби побудови віртуальних приватних мереж тощо. До нього відносяться майже всі наявні на ринку брандмауери.

1.4 Використання міжмережевого екрану на прикладі апаратного брандмауеру

Сучасні мережеві атаки часто використовують кілька складних методів проникнення в корпоративні мережі. Щоб уникнути виявлення систем запобігання вторгненням, ці атаки часто кодуються за допомогою складних алгоритмів. Після захоплення мети під контроль зловмисник спробує завантажити та встановити шкідливе ПЗ на "захоплене" пристрій. У багатьох випадках використовуються шкідливі програми нових версій, які можуть виявити традиційні антивірусні рішення. Більше того, маючи істотні трудові та грошові

ресурси, розробники шкідливих рішень підлаштовуються під присутні на ринку рішення щодо захисту мереж, принцип їх роботи та аналізу трафіку. Наприклад, сучасні атаки використовують SSL-шифрування, щоб приховати завантаження шкідливого програмного забезпечення або навіть замаскувати трафік командного управління, переданого атакуючим. Або ж, знаючи про обмеженість у можливостях міжмережевих екранів попереднього покоління обробляти великі пакети у значній кількості, маскують шкідливий код у великому трафіку.

Всі продукти Dell SonicWALL NGFW від молодших до старших моделей використовують запатентовану технологію однопрохідного двигуна з малим часом затримки Reassembly-Free Deep Packet Inspection (RFDPI), який перевіряє кожен байт кожного пакета, зберігаючи при цьому високу швидкість передачі і продуктивність. Продукти SonicWALL для захисту мережі від внутрішніх та зовнішніх атак, що використовують вразливість програм, сканують весь трафік, незалежно від розміру файлів, порту або протоколу. Можливості візуалізації та контролю використовуються для вивчення кожного пакета, щоб визначити, які додатки використовуються, і хто їх використовує. Принцип роботи технології RFPDI - розбиття вхідного пакета на сегменти та паралельне, одночасне сканування кожного сегмента. Дозволяє це реалізувати багатоядерна архітектура процесорів, використовуються у міжмережевих екранах SonicWALL. Природно, що технологія RFPDI інтегрована з платформою самого пристрою і дозволяє оптимізувати управління деталізованими політиками міжмережевого екрану як через інтерфейс міжмережевого екрану, так і через систему Dell SonicWALL Global Management System.

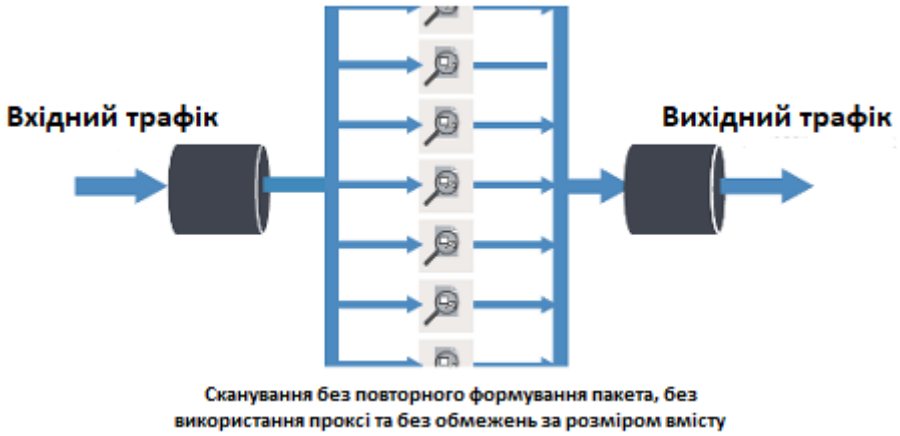

Рис. 1.1. Схема сканування пакетів

Розшифровка та перевірка SSL-трафіку є одним із головних та обов'язковим елементом забезпечення більш глибокого рівня мережної безпеки у мережах нового покоління. Аналітики стверджують, що майже 35 відсотків корпоративного мережевого трафіку шифрується за допомогою SSL. Таким чином, організації, які не мають відповідного обладнання та не перевіряють SSL-трафік, фактично залишають без перегляду одну третину трафіку в мережі, що призвело до підвищеної уваги зловмисників саме до SSL, так як гарантує стовідсотковий успіх атак таких організацій. З цього випливає, що організація повинна мати можливість перевіряти весь трафік, і на будь-якому порту, незалежно - із шифруванням SSL він чи ні. Відповідну можливість надають міжмережевий екран Dell SonicWALL, перевіряючи шифрований SSL і нешифрований трафік на кожному порту.

Крім того, міжмережеві екрани Dell SonicWALL використовують фірмовий ресурс мережі ідентифікації атак та моніторингу мережі Dell SonicWALL Global Response Intelligent Defense (GRID). Мережа є понад 1 мільйон розкиданих по всій планеті, як їх називає виробник, "сенсорів". Ці сенсори збирають дані про підозрілу мережеву активність, шкідливе ПЗ, атаки і відправляють в умовний "Центр", де дані аналізуються на предмет реальності загрози, випускаються оновлення баз і сигнатур, після чого розсилаються всім пристроям SonicWALL на планеті.

Окремим питанням є оновлення баз та засобів мережевого захисту від шкідливого ПЗ. Нам логічно зрозуміло, що якщо зловмисник випускає засіб, який використовує вразливість у будь-якому ПЗ, то у зловмисника є карт-бланш на "збирання врожаю" на цій уразливості доти, доки:

1. Виробник уразливого ПЗ не зробить латку-патч та/або виробник захисної системи не оновить бази та засоби, що перешкоджають використанню вразливості;

2. Засоби, що забезпечують захист, не отримають оновлення.

В цілому, NGFW від Dell SonicWALL забезпечує інтеграцію засобів мережевої безпеки, веб-захисту та безпеки електронної пошти. Це багаторівневий захист від вірусів, хробаків, троянів, шпигунського ПЗ та вторгнень. Веб-захист пристроїв забезпечує зручне управління блокуванням сайтів, що не відповідають вимогам, та контроль використання миттєвого обміну повідомленнями та піринговых додатків.

Рішення Dell SonicWALL для аналізу, управління та візуалізації програм розширюють контроль над невиробничими програмами, наприклад, сервісами онлайн-торгівлі, обміну миттєвими повідомленнями/чату, пірингового обміну та потокової передачі відео. Рішення Dell SonicWALL для захисту електронної пошти забезпечують захист від спаму та фішинг-атак, щоб позбавити співробітників від отримання шахрайських та неправомочних електронних повідомлень. Інтелектуальні рішення Dell SonicWALL спрощують та знижують вартість централізованого управління локальними, віддаленими та мобільними мережевими службами, захищаючи ключову інформацію та комунікаційні ресурси.

Міжмережеві екрани Dell SonicWALL надають можливість використовувати 3G/4G-зв'язок, що за невеликі гроші забезпечує організації принципово новий рівень резервування Інтернет-доступу за альтернативним, виділеною лінією, фізичним способом передачі даних через бездротовий зв'язок, що в 00-х роках концептуально було доступно тільки компаніям із великими IT-бюджетами рахунок організації каналів супутникового зв'язку. До того ж,

міжмережеві екрани Dell SonicWALL дозволяють мати кілька одночасно підключених WAN з'єднань і не тільки перекидати трафік з каналу на канал у разі відмови робочого з'єднання, але й використовувати канали одночасно, балансуючи реальний трафік між ними на рівні додатків, відправляючи, наприклад, критичний трафік на більш надійні та дорогі канали, а другорядний - на менш надійні та дешевші.

Для контролю програм у SonicWALL використовується функція Application Intelligence and Control, яка забезпечує детальний контроль та візуалізацію програм у режимі реального часу, гарантуючи пріоритизація пропускної спроможності. Ця функція використовує вже згадану запатентовану технологію Reassembly-Free Deep Packet Inspection (RFDPI) для розпізнавання та контролю програм незалежно від порту або протоколу. Зараз база сигнатур розпізнає понад 4600 додатків та мільйони видів шкідливого ПЗ, продовжуючи розширюватися та забезпечувати розподіл смуги пропускання та забороняти доступ до веб-сайтів. Подивитися в режимі реального часу за використанням програм можна за допомогою функції Application Flow Monitor, яка дає відомості про вхідну та вихідну смуги пропускання,

Технологія Dell SonicWALL Clean VPN призначена для забезпечення безпечного доступу та взаємодії з віддаленими користувачами при з'єднаннях IPSec та SSL VPN. Сам VPN доступ захищається через автентифікацію, шифрування даних та налаштування доступу. При цьому перевіряється одночасно як вхідний, так і вихідний трафік VPN. Перед попаданням трафіку в мережу всередині "периметра" весь VPN трафік дешифрується і очищається. На додаток, за допомогою функцій Application Intelligence and Control є можливість візуалізації трафіку в тунелях VPN і застосування політик з контролю смуги пропускання, пріорітізуючи трафік потрібних додатків і блокуючи (або обмежуючи) трафік непотрібних.

Список протоколів, аналізованих системою запобігання вторгненню Dell SonicWALL:

- IPv4/IPv6/SSL

- TCP/ICMP/DNS
- HTTP/HTTPS
- SMTP/IMAP/POP3
- FTP/FTPS
- Telnet/SNMP
- SIP/H.323
- RTP/RPC
- MySQL/MS-SQL
- NetBIOS/SMB/SMB2

1.5 Як працює міжмережевий екран – Netfilter в Linux

У Linux брандмауер вбудований в ядро і називається Netfilter. А для керування ним можуть використовуватися різні утиліти. У Debian та Ubuntu більш старших версій використовувалася утиліта iptables. А в Debian 11 та Ubuntu 22.04 почали використовувати – nftables. Саме з цією утилітою я вас і познайомлю. Але спочатку дізнаємося, як взагалі працює Netfilter.

Фаєрвол Netfilter ділить весь трафік на ланцюжки (chain) і обробляє кожен ланцюжок окремо. (Рис 1.2)

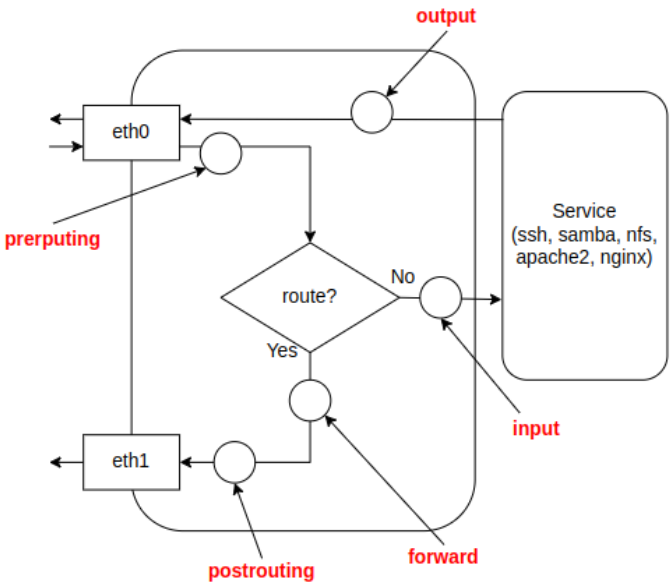

Рис. 1.2. Проходження пакетів по ланцюжках Netfilter в OS Linux

Вхідний трафік відразу обробляється в ланцюжку prerouting (тут може спрацювати вхідний NAT). Після чого, якщо трафік:

- призначений серверу, що потрапляє в ланцюжок input (тут пакети можна прийняти або відкинути).
- не призначений серверу (проходить крізь нього), то потрапляє в ланцюжок forward (тут також пакети можна прийняти або відкинути).

Якщо трафік потрапив у ланцюжок forward, то перед виходом він потрапить у ланцюжок postrouting (тут може спрацювати вихідний NAT). А весь вихідний трафік обробляється ланцюжком output (тут також може відпрацювати вихідний NAT).

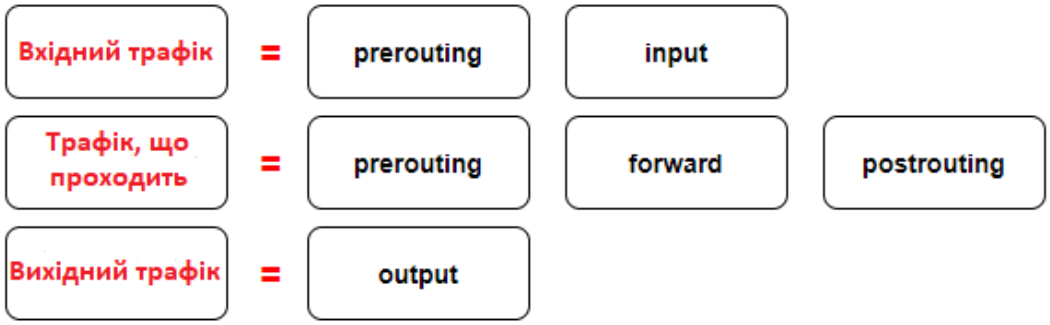

Рис. 1.3. Напрямок трафіку та проходження пакетами ланцюжків

Над пакетами здійснюються різні дії, що групуються у таблиці. І кожен ланцюжок може входити в ту чи іншу таблицю, або може перебувати в кількох або всіх таблицях відразу. Якщо ланцюжок входить у певну таблицю, це означає, що в цьому ланцюжку над пакетами можна виконувати дії з цієї таблиці. Список таблиць:

1) **Filter** – тут пакети фільтруються, тобто їх можна пропустити чи відкинути. Містить ланцюжки: input, forward, output.
2) **Nat** – тут працює NAT (прокидання портів, вхідний та вихідний NAT). Містить ланцюжки: prerouting, output, postrouting.
3) **Mangle** – дозволяє вносити зміни до заголовків ip-пакетів. Містить усі ланцюжки.
4) **Raw** – дозволяє вибірково пропускати або відкидати пакети перед тим, як вони потраплять до механізму відстеження з'єднання (connection tracking), що значно знижує навантаження на процесор. Містить ланцюжки: prerouting та output.

1.6 Файрвол або фільтрація пакетів

Файрвол - це комп'ютерне обладнання, яке може бути представлене у вигляді програмного, апаратного або програмно-апаратного комплексу. Файрвол обробляє вхідні та вихідні мережеві пакети (що надходять у локальну мережу чи покидають її) і пропускає лише ті, які відповідають деяким зумовленим правилам.

Фільтруючий мережевий шлюз - це різновид файрвола, який захищає всю мережу. Зазвичай його встановлюють на виділеному комп'ютері, налаштованому як мережний шлюз. В результаті, цей комп'ютер може обробляти всі пакети, які перетинають кордон між мережею і зовнішнім світом. Як варіант, локальний файрвол — це програмна служба, яка працює на конкретній машині для того, щоб обмежувати доступ до деяких служб на цій машині, або, можливо, для запобігання вихідним з'єднанням, які ініціюються небажаним програмним забезпеченням, яке користувач комп'ютера може навмисно або випадково встановити.

У ядрі Linux є вбудований файрвол netfilter. Вимоги різних мереж та користувачів неоднакові, тому немає стандартного підходу до налаштування файрвола, що дозволяє отримати готове рішення на всі випадки життя. Керувати файрволом netfilter можна з простору користувача за допомогою командiptablesiip6tables. Різниця між цими двома командами полягає в тому, що перша призначена для мереж IPv4, друга – для мереж IPv6. Так як обидва стеки протоколів, ймовірно, будуть в ходу ще багато років, обидва інструменти слід використовувати спільно. Крім того, тут можна застосувати чудову програму fwbuilder яка дає графічні інструменти для побудови та подання правил фільтрації трафіку. Як би ви не вирішили налаштовувати netfilter - це стандартний Linux файрвол, тому поглянемо ближче на те, як він працює.

Netfilter використовує чотири таблиці, які зберігають правила, що регулюють три види операцій над пакетами:

1) filter: стосується правил фільтрації (прийняти, відхилити або проігнорувати пакет)

2) nat(Network Address Translation): стосується трансляції адрес джерела, одержувача, або портів пакета
3) mangle: стосується інших змін IP-пакетів (включаючи поля та параметри ToS - Type of Service)
4) raw: дозволяє виконувати ручні модифікації пакетів, перш ніж вони дійшли до системи відстеження з'єднання.

Кожна таблиця містить списки правил, які називаються ланцюжками. Файрвол використовує стандартні ланцюжки для обробки пакетів, ґрунтуючись на певних умовах. Адміністратор може створювати інші ланцюжки, які будуть використані тільки у випадках, коли на них, прямо чи опосередковано, посилається один із стандартних ланцюжків.

Таблиця filter містить три стандартні ланцюжки:

1) INPUT: стосується пакетів, пунктом призначення яких є сам файрвол;
2) OUTPUT: стосується пакетів, які надсилає файрвол;
3) FORWARD: стосується пакетів, що проходять через файрвол (який не є ні їхнім джерелом, ні пунктом їх призначення).

Таблиця NAT теж містить три стандартні ланцюжки:

1) PREROUTING: для модифікації пакетів, що прибувають
2) POSTROUTING: для модифікації пакетів, коли вони готові вирушити до пункту призначення
3) OUTPUT: для модифікації пакетів, що генеруються самим файрволом

Вищеописані ланцюжки показані нижче на таблиці (Рис 1.4)

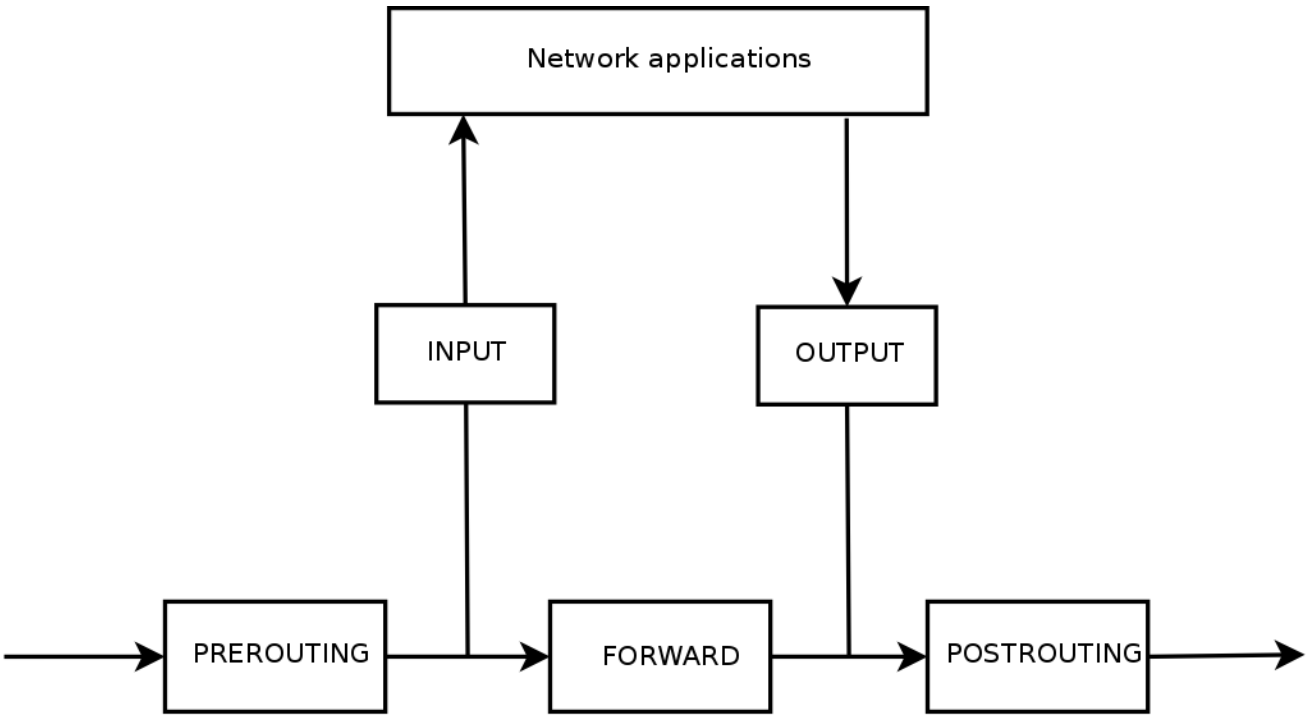

Рис. 1.4. Ланцюжки Netfilter

Кожен ланцюжок є список правил. Кожне правило – це набір умов та дій, які мають бути виконані за дотримання відповідних умов. У процесі обробки пакета файрвол переглядає відповідний ланцюжок, правило за правилом, і, коли умови для певного правила виконуються, він «перестрибує» (звідси й опція-j -від слова "jump") до заданої дії для продовження обробки пакета.

2 АНАЛІЗ РОБОЧОГО ПРОСТОРУ ТА СТВОРЕННЯ ВІРТУАЛЬНОГО СЕРЕДОВИЩА

2.1 Аналіз робочого простору віртуальної мережі

Наша мета в цій частині – написання програми, яка на дуже простому рівні контролюватиме весь трафік. А саме – ми визначимо, які пакети можна пропускати, а які видаляти. Створимо просту систему логів, для відстеження результатів, а так само програму для користувача, через яку можна буде читати ці результати і трохи керувати програмою. Такий ось тривіальний firewall або російською мовою - Міжмережевий екран, мережевий екран - це комплекс апаратних і програмних засобів в комп'ютерній мережі, що здійснює контроль і фільтрацію проходять через не гістьових пакетів відповідно до заданих правил. У нашому випадку, функцію firewall виконуватиме окремий комп'ютер (Рис. 2.1).

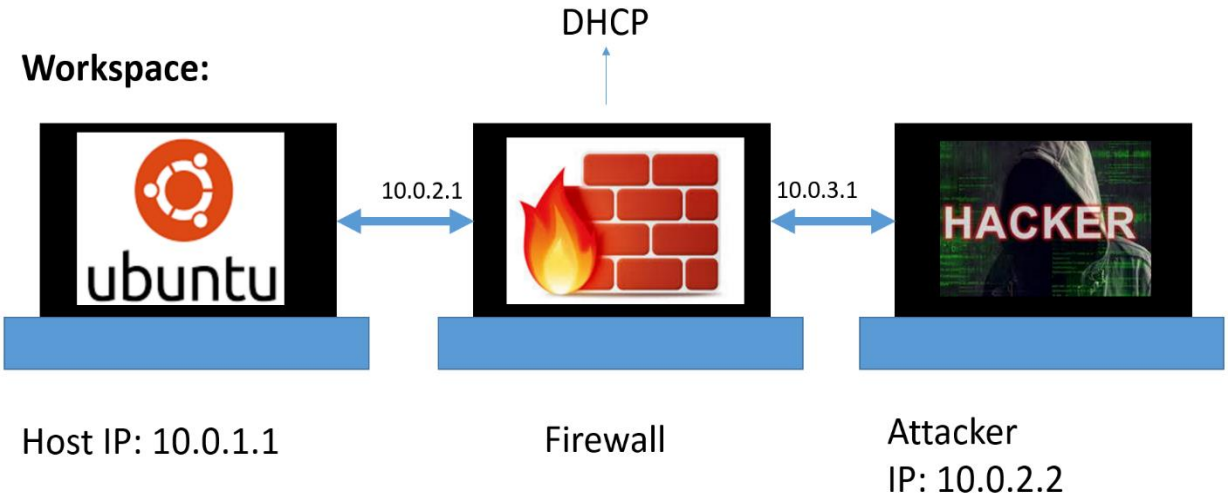

Рис. 2.1 З'єднана віртуальна мережа

Ми зробимо мережу, в якій буде три віртуальні комп'ютери – host1, host2, fw і з'єднаємо їх як на зображенні вище.

host1 – отримає статичний IP address – 10.0.1.1. Його ми захищатимемо.

host2 - який буде в останній частині атакуючим і матиме статичний IP - 10.0.2.2

fw - відстежуватиме трафік, сконфігуруємо йому так само інтерфейс для виходу в інтернет, щоб можна було при необхідності завантажувати потрібний софт.

	host1	fw	host2
ip	10.0.1.1, static	10.0.1.3, static 10.0.2.3, static DHCP, dynamic	10.0.2.2, static
forward	10.0.1.3	-	10.0.2.3

Рис 2.2 Схема робочого простору

У двох словах про DHCP (англ. Dynamic Host Configuration Protocol - протокол динамічного налаштування вузла) - мережевий протокол, що дозволяє комп'ютерам автоматично отримувати IP-адресу та інші параметри. роботи в мережі TCP/IP.

Тобто він сам отримуватиме різні налаштування, включаючи IP, для того щоб мати можливість через цей мережевий інтерфейс, виходити без зайвого головного болю в інтернет.

2.2 Створення віртуальних середовищ

Трохи про віртуальні машини. У нашому випадку, віртуальна машина – це програма, яка емулюватиме комп'ютер, завдяки чому ми зможемо запустити на нашому комп'ютері відразу три операційні системи.

Ми скористаємось безкоштовною програмою VirtualBox, яка дозволяє створювати віртуальні мережі на одному комп'ютері. Звичайно ж не важливо, чим саме користуватися, головне мати можливість сконфігурувати мережу. Програма безкоштовна, шукаємо, качаємо, ставимо.

Наступним кроком додасть віртуальні машини. Я працюватиму з Linux Ubuntu 12, з «історичних причин». Ви можете завантажити останню версію. Тим більше, якщо ваш комп'ютер досить потужний.

Тепер у нас є щось схоже на картинку. Я раджу скористатися функцією “clone” у virtualbox для швидкого створення ще двох машин. У мене в ролі host1,

host2 спеціально урізані легкі версії ubuntu, без графічного інтерфейсу. Приблизно так це має виглядати наприкінці (Рис. 2.3)

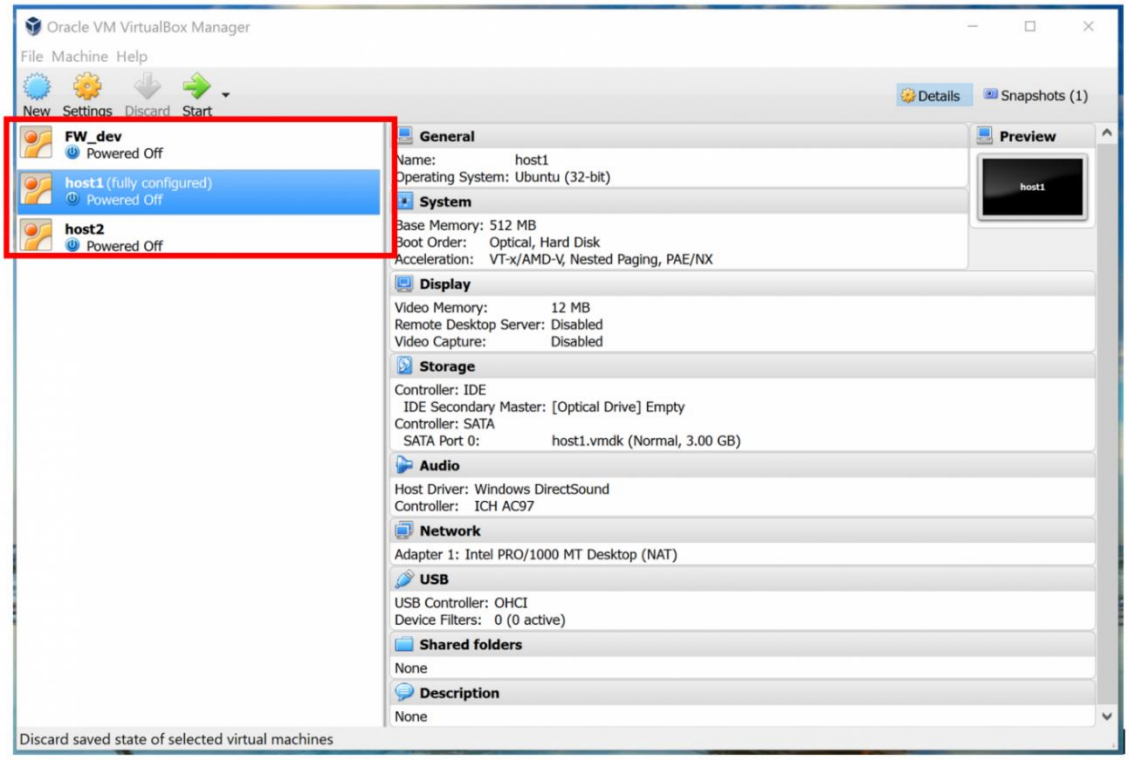

Рис 2.3 Створені віртуальні машини за допомогою VirtualBox

У нас вже є три віртуальні linux машини, які ми зараз налаштуємо в одну мережу і перевіримо, що все працює, перш ніж почати писати програму. Ідемо у settings та конфігуруємо як на рисунку (Рис. 2.4):

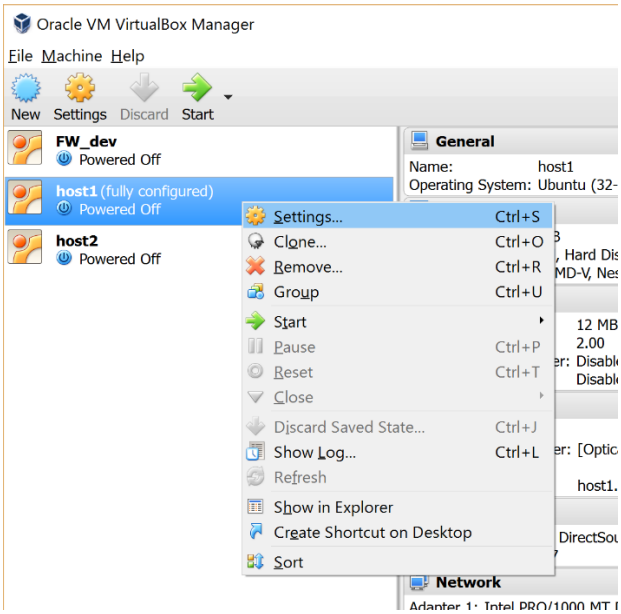

Рис. 2.4 Налаштування віртуальної машини

Потім додаємо мережевий пристрій (Рис. 2.5):

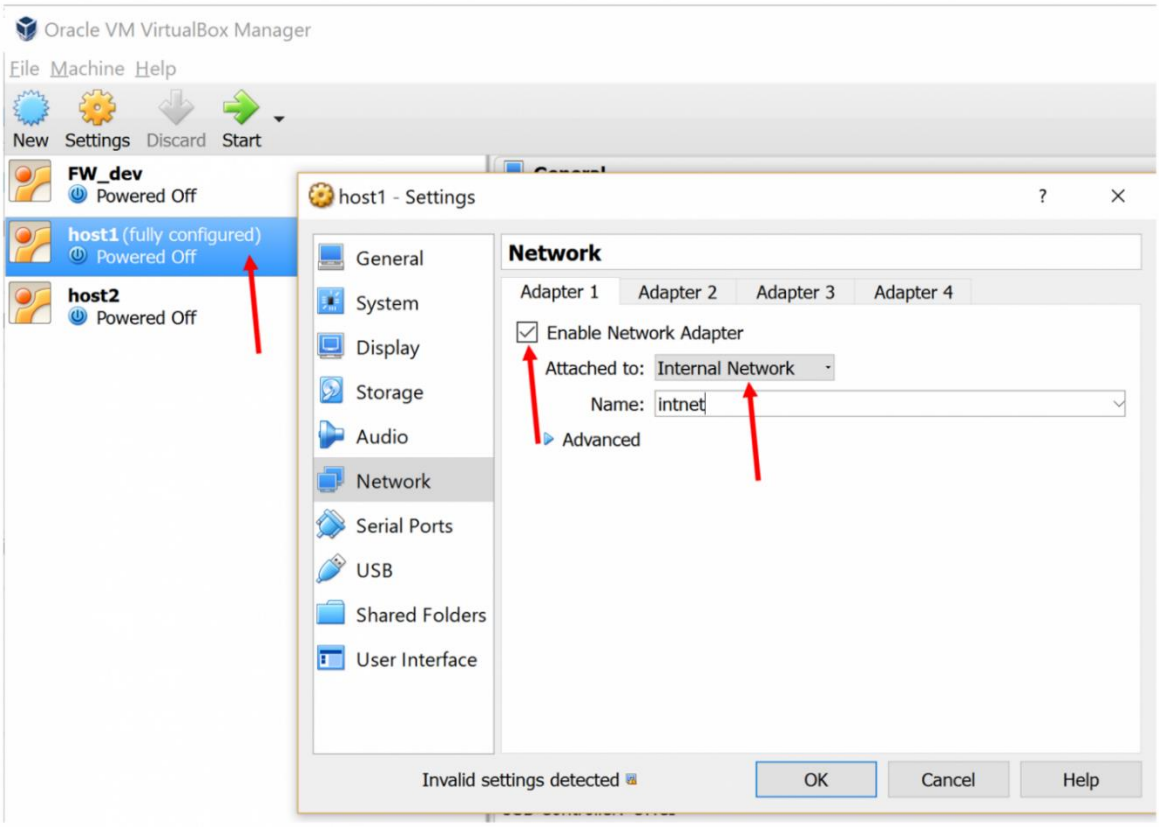

Рис. 2.5 Додавання мережевого пристрою

Тепер, linux “думатиме”, що в нашого комп'ютера, є мережна карта, а якщо ви подивитесь в Advanced, то побачите, що в неї ще й вставлений кабель. Чудово. Тепер треба налаштувати таку карту в операційній системі. Для цього запускаємо host1. Я редагуватиму vi, але не забуваємо давати права адміністратора (запускати з sudo), щоб мати можливість зберегти. Редагуємо даний файл через vi /etc/network/interfaces (Рис. 2.6).

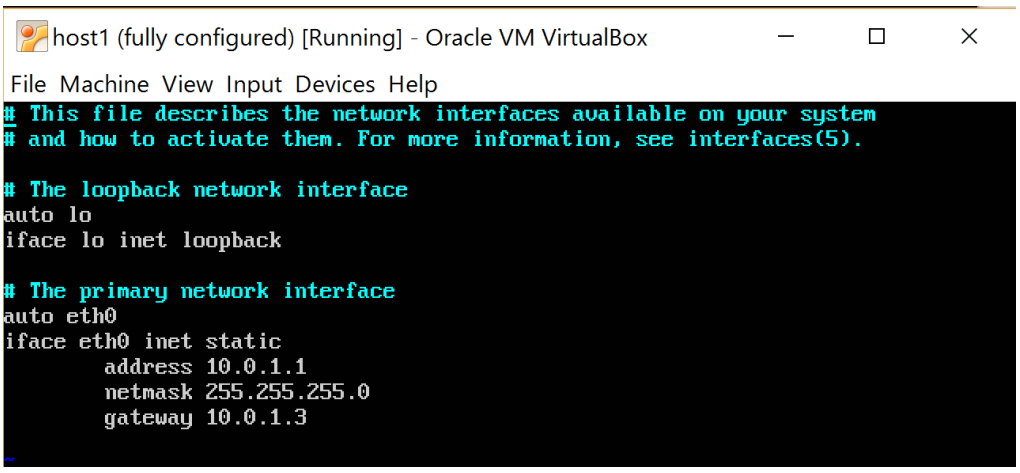

Рис. 2.6 Редагування /etc/network/interfaces

Запускаємо ifconfig та перевіряємо, що у нас є мережева карта під ім'ям eth0, з IP = 10.0.1.1, маскою і т.д. (Рис. 2.7)

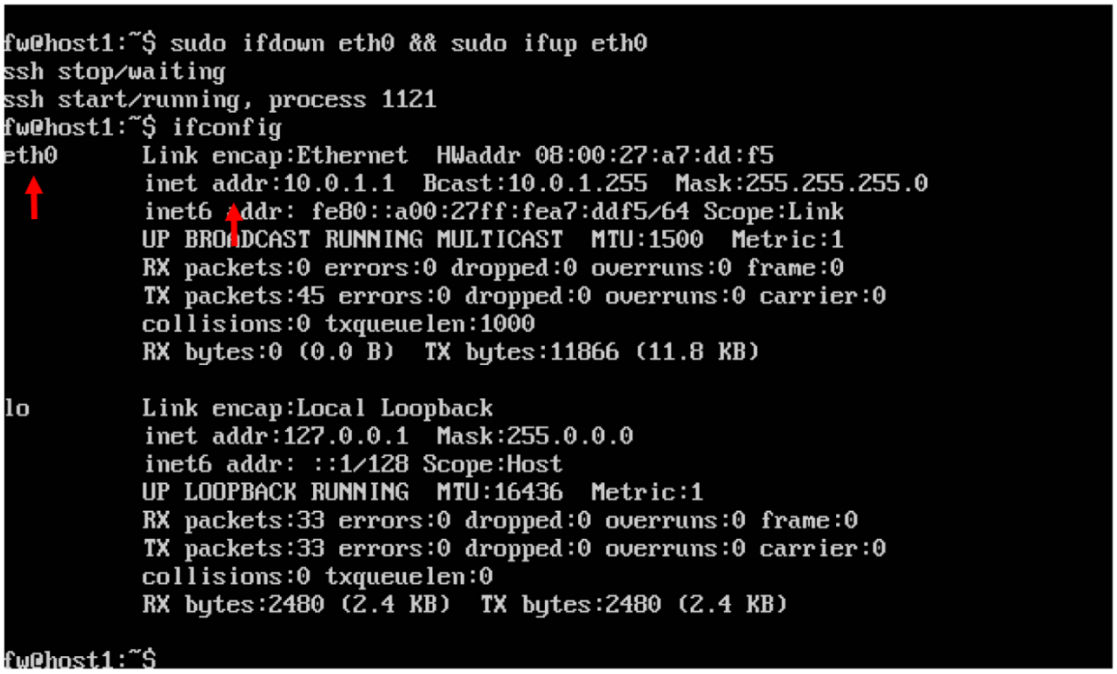

Рис. 2.7 Інформація мережевих інтерфейсів ifconfig

Те ж саме робимо з host2, тільки IP встановити в 10.0.2.2, gateway 10.0.2.3 і не забути “додати” мережну карту в налаштуваннях машини у virtualbox (у мене після змін налаштувань у virtualbox машина при завантаженні підвисала на пару хвилин і писала waiting for network configuration… Це пов'язано з тим, що за умовчанням, мережний інтерфейс налаштований як dhcp, тобто, операційна система чекає динамічно отримати мережеві налаштування, але в virtualbox, для цього вона має бути налаштована як NAT, але ми це змінили, тому ОС намагається зрозуміти, що не так, можна спочатку завантажити ОС, відредагувати interfaces, потім вимкнути комп'ютер, змінити налаштування і запустити заново) (Рис. 2.8).

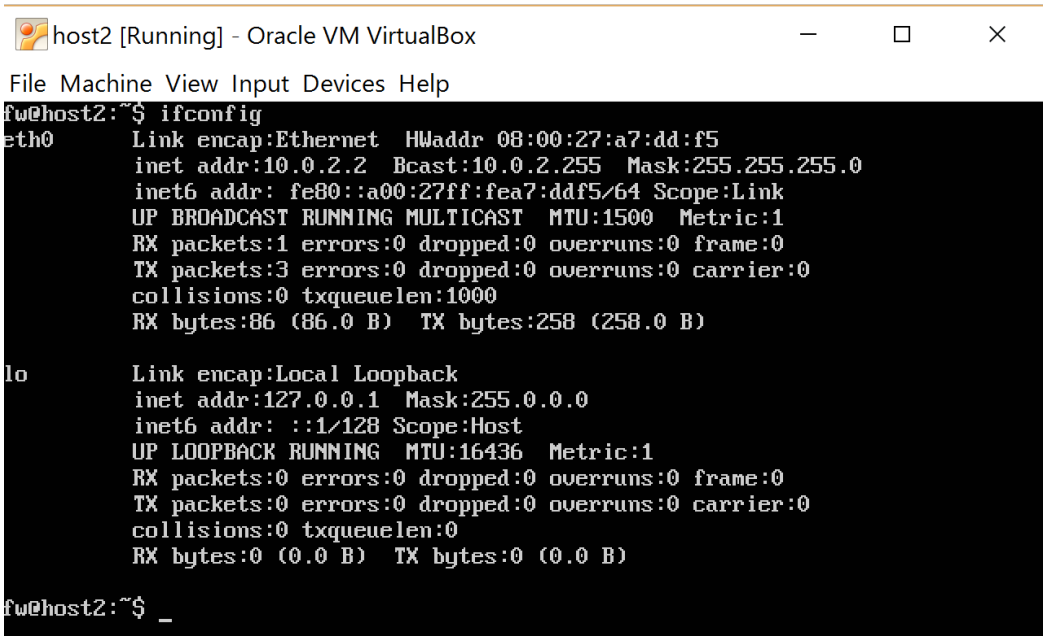

Рис. 2.8 Мережеві інтерфейси Host2

Тепер настала черга головної машини, яка виконуватиме роль firewall. Нагадую, що нам знадобляться три мережеві карти, через одну ми зможемо виходити в інтернет у разі потреби, тому вона буде визначена як NAT. Інші дві будуть відповідальні пересилати трафік з host1 > host2 і назад з host2 > host1. (Рис. 2.9)

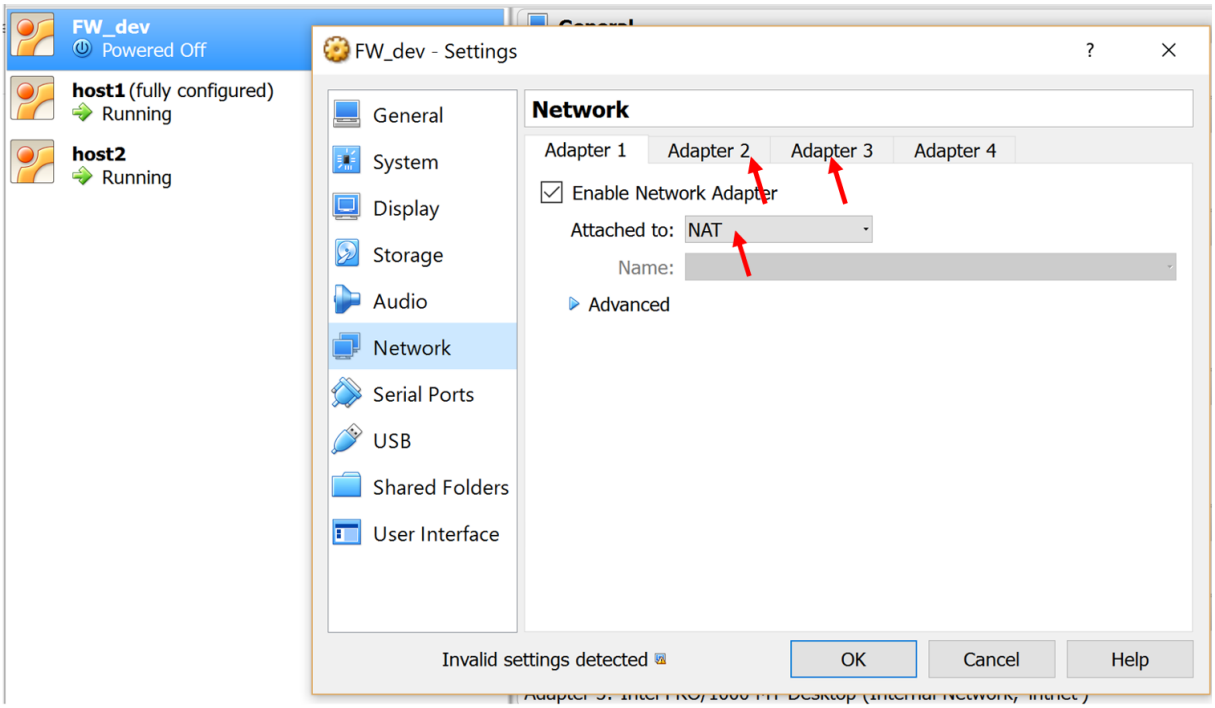

Рис. 2.9 Визначення Firewall машини як NAT

Adapter1 – має бути виставлений як NAT. Налаштування Adapter2, Adapter3, такі ж як і попередніх прикладах (Internal network). Запускаємо, починаємо конфігурувати мережеві карти. Так має виглядати файл interfaces. (Рис. 2.10)

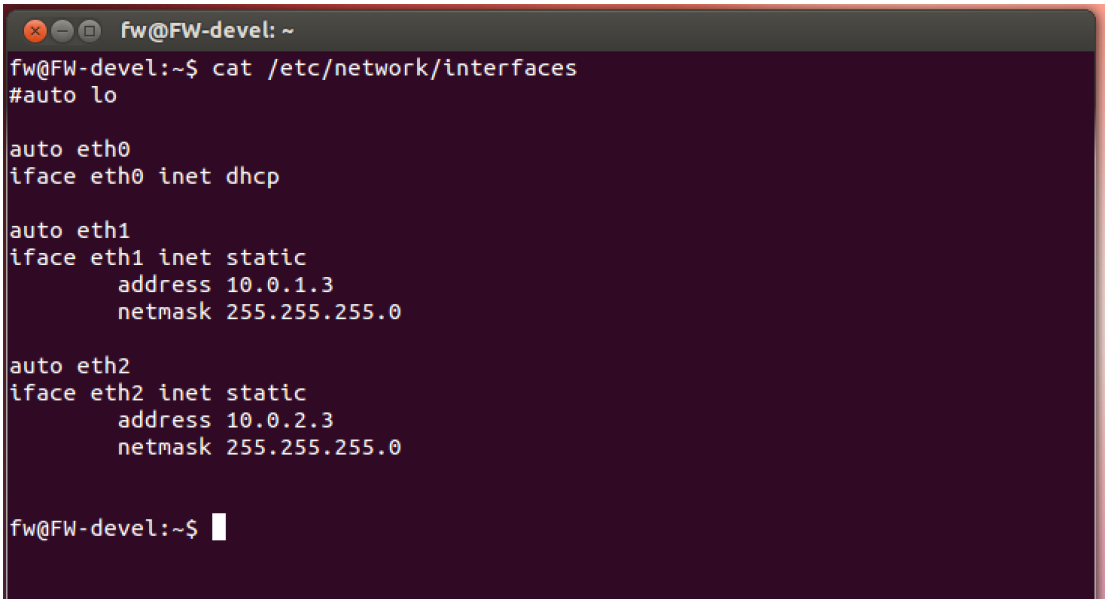

Рис. 2.10 Вигляд файлу interfaces

А ось – кінцевий результат конфігурації після перезапуску:

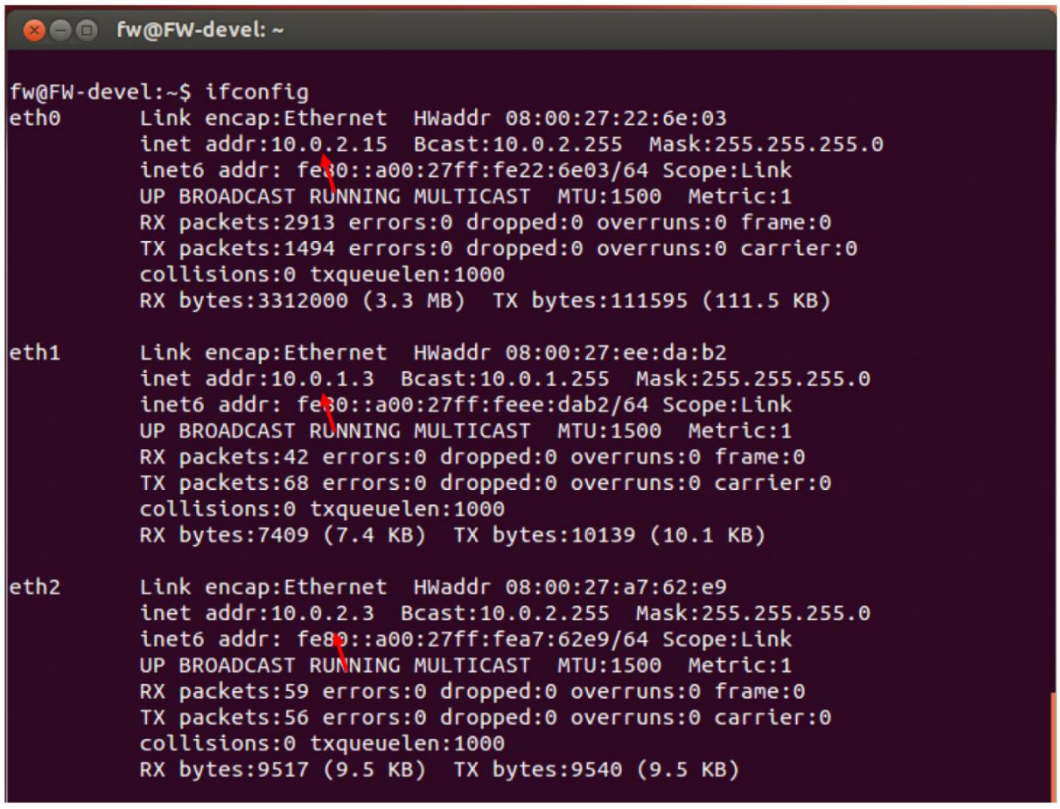

Рис 2.11 Вигляд ifconfig після перезапуску

Останнє, що залишилося зробити - це дозволити packet forwarding, для того щоб комп'ютер міг брати пакети з одного мережного інтерфейсу (10.0.2.3), на який приходять пакети від host2 і передати їх на інший мережний інтерфейс 10.0.1.3, зв'язаний відповідно з host1. Щоб налаштування було постійним, потрібно змінити наступний рядок /etc/sysctl.conf як net.ipv4.ip_forward = 1. Залишилося перевірити, що все складно одне з одним і працює. Перевіряти будемо класичним ping (Рис. 2.12)

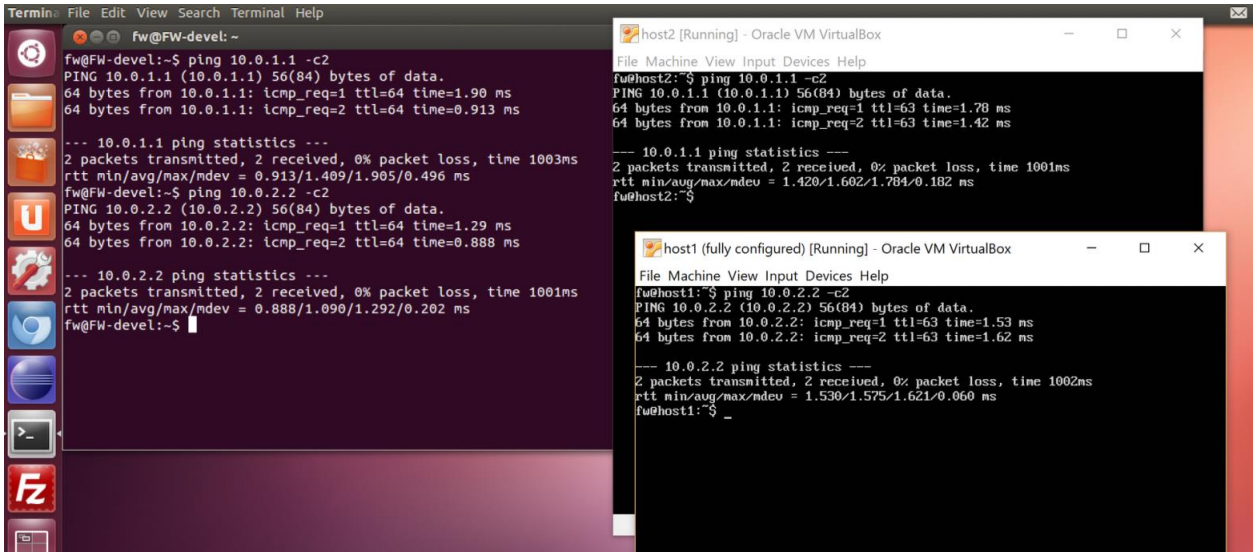

Рис. 2.12 Команда ping між віртуальними машинами

Як видно на картинці, ping “пройшов” з host1 > host2, host2 > host1, fw > host1, fw > host2. Готово “лабораторія” для майбутніх експериментів налаштована.

3 ЗАСТОСУВАННЯ ТЕХНОЛОГІЇ БЕЗПЕКИ КОРПОРАТИВНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1 Встановлення лабораторного стенду

Всі говорять, що для захисту мережі потрібно застосовувати міжмережевий екран, але ніхто не говорить, як це потрібно робити. Що ж, виправимо ситуацію, розглянемо типові сценарії застосування міжмережевих екранів і те, як їх у своїй налаштовувати. Як міжмережевий екран будемо використовувати nftables, що функціонує під керуванням ОС Debian GNU Linux. Тут ми не розглядатимемо принципи роботи nftables, синтаксис його командного рядка або формат конфігураційних файлів. Ми зосередимося на його практичному використанні. За великим рахунком, всі необхідні налаштування nftables будуть докладно описані, так що проблем з їх відтворенням, по ідеї, бути не повинно.

Для побудови лабораторних стендів можна використовувати будь-яку зручну вам систему віртуалізації: VMware, Hyper-V, VirtualBox, QEMU-KVM або іншу. Головною вимогою до цієї системи буде можливість управління віртуальними мережами. Система має дозволяти створювати віртуальні мережі, і навіть призначати їх у різні адаптери віртуальних машин.

У більшості лабораторних стендів нам знадобиться віртуальна машина-шаблон, яку ми тиражуватимемо і доналаштовуватимемо. Для створення цього шаблону потрібно:

1. Завантажити Debian 11.4 у форматі netinstall.
2. Встановити ОС у мінімальній конфігурації. Для цього під час інсталяції вибираємо всі параметри за замовчуванням (Next, Next, Next, ...), а на кроці з вибору пакетів "Software selection" відмовляємось від усього пропонованого (Рис 3.1).

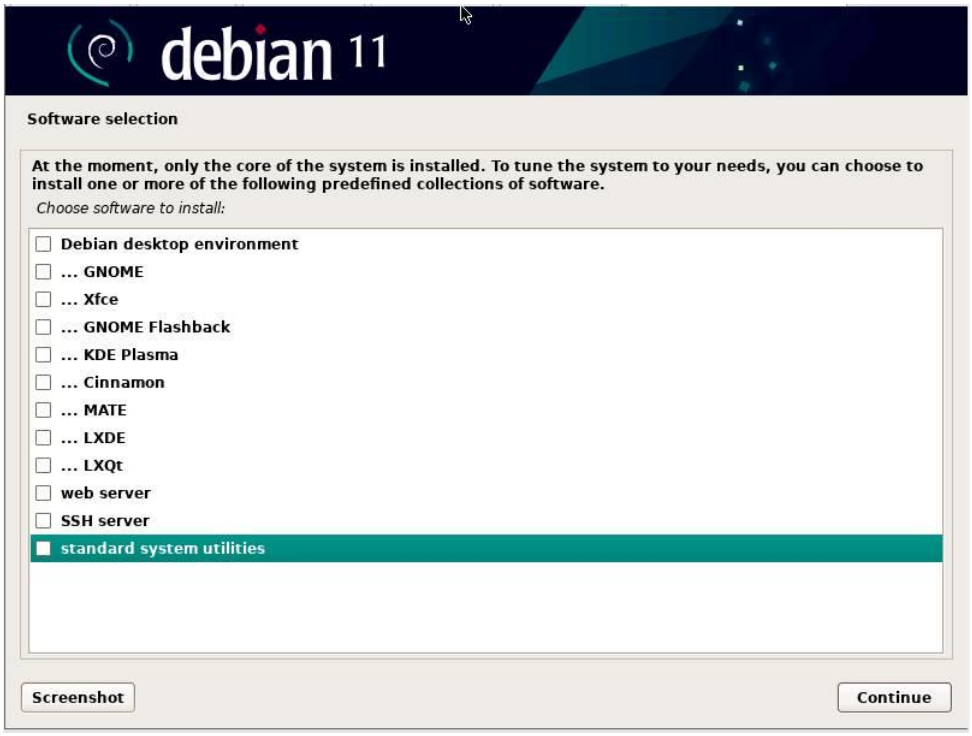

Рис. 3.1 Інсталяція Debian

3. Рекомендується встановити пакет розширень віртуалізації для гостьової операційної системи. Наприклад, для VMware – це буде VMware Tools, VirtualBox – Virtualbox extension pack тощо. Після цього рекомендується налаштувати каталог, що розділяється (Shared Folder) для зв'язку між файловими системами віртуальної машини і гіпервізора.
4. Для підвищення зручності виконання практичних робіт рекомендується поставити додаткові пакети:

    а. OpenSSH-сервер (пакет ssh).

    б. Файловий менеджер Midnight Commander (пакет MC).

    в. Утиліту conntrack, що демонструє внутрішній стан брандмауера та допомагає у налагодженні правил фільтрації (пакет conntrack).

Усі ці пакети можна легко поставити. Для цього достатньо зайти на машину під root'ом (при конфігуруванні Linux нам завжди буде потрібно root) і виконати в консолі команду apt install mc ssh conntrack.

3.2 Захист робочої станції за допомогою локального брандмауера та розгляд моделі загроз

Завдання налаштувати локальний брандмауер - мабуть, найпоширеніша задача міжмережевого екранування. Для її вирішення ми повинні знати відповіді на два питання: чого ми хочемо досягти і як це реалізувати.

Відповідь на перше запитання має дати архітектор інформаційної безпеки, він має сформулювати загрози та запропонувати методи їхньої нейтралізації. Проблема в тому, що в невеликих колективах людей, які грають таку роль, немає і всім цим доводиться займатися системним адміністраторам. Незважаючи на це, все ж таки рекомендується діяти за стандартним алгоритмом, тобто так, як би діяв архітектор.

Даний алгоритм може здатися забюрократизованим, однак це не зовсім так, адже більшість кроків потрібно буде зробити лише раз, а потім виконувати роботи за вже сформованими шаблонами. Безперечним плюсом стандартного алгоритму є забезпечення доказового рівня безпеки, при якому всі заходи захисту, що вживаються, чітко обґрунтовані і можуть бути легко верифіковані. Все разом це підвищує рівень зрілості компанії щодо забезпечення інформаційної безпеки і несе безліч супутніх фішок. Стандартний алгоритм розв'язання задач міжмережевого екранування:

1. Формулювання моделі загроз.
2. Аналіз загроз та вироблення заходів щодо їх нейтралізації. На даному етапі необхідно сформулювати схему розмежування трафіку: тобто, який трафік пропускаємо, а який блокуємо.
3. На підставі схеми розмежування трафіку здійснюється налаштування міжмережевого екрану: формуються правила фільтрації, налаштовуються сервіси, пишуться скрипти і таке інше.

Моделювання загроз – процес нетривіальний і потребує значних компетенцій у сфері інформаційної безпеки. Фактично необхідно знати те, як вас намагатимуться зламати. Безліч способів атак вже описано, але постійно з'являються нові й нові. Фахівці, які професійно займаються моделюванням загроз, повинні постійно тримати руку на пульсі та бути в курсі всіх модних новинок. Для всіх інших істотною підмогою будуть готові моделі та банки даних загроз. Стосовно нашого завдання ми розглядатимемо такі загрози:

Загроза	**Коментар**
У1. Загроза віддаленої експлуатації уразливостей (як програмних, так і конфігураційних) у мережевих сервісах вузла.	Приклади реалізації загрози: епідемія хробака MSBlast (експлуатація програмної вразливості) або множинні витоку даних з кластерів Elastic search, що відбуваються через відсутність авторизації (експлуатація конфігураційної вразливості).
У2. Загроза віддаленої атаки на відмову в обслуговуванні (DoS) мережевих сервісів вузла.	
У2.1. DoS атака за рахунок перевищення критичної кількості мережевих запитів, які може обробити мережевий сервіс з прийнятною якістю	Подібні атаки найбільш актуальні для серверів баз даних, які можуть генерувати суттєве обчислювальне навантаження на обробку запиту, що входить.
У2.2. DoS атака за рахунок підробки зловмисником IP-пакету та встановлення в ньому зворотної адреси рівною адресою петлі 127.0.0.0/8	Мережевий сервіс, отримуючи подібний запит, при надсиланні відповіді може отримати його назад, що може призвести до зациклювання або інших негативних наслідків.

У3. Загроза розкриття автентифікаційних даних за рахунок віддалених атак прямого перебору, які здійснюються щодо мережевих сервісів вузла.	Тривіальний підбір паролів, але здійснюваний віддалено через мережу.
У4.Загроза встановлення вихідних шкідливих мережних з'єднань локальним програмним забезпеченням вузла.	Шкідливе мережне з'єднання може встановити як троян, що зв'язується зі своїм сервером управління (C&C), так і легітимна програма, що передає надлишкові дані (наприклад, телеметрію) на сервери розробників. Крім того, до цієї загрози можна віднести і дії співробітників, які використовують робочі комп'ютери особистих потреб (наприклад, для майнінгу криптовалют).
У5.Загроза несанкціонованого відкриття мережного порту шкідливим кодом.	На зорі вірусобудування перші шпигунські шкідливі програми відкривали на машині жертви мережеві порти, до яких потім підключалися особи, які займаються шпигунством.

Таб. 3.1 Моделі загроз

Тепер проаналізуємо загрози та сформулюємо ідеї щодо їх нейтралізації. Для парирування загрози У1 та У5 необхідно позбавити зловмисників можливості здійснювати підключення до мережевих сервісів вузла. Це досягається шляхом обмеження вхідного трафіку за портами та адресами джерела (білі або чорні списки).

Нейтралізація У2.1 базується на обмеженні швидкості отримання сервісом мережевих пакетів, а як складніший варіант - виявлення атакуючих вузлів та блокування їх IP. Слід зазначити, що обмеження максимальної частоти підключень який завжди припустимо, особливо для громадських сервісів.

Загроза У2.2. нейтралізується шляхом відкидання всіх мережевих пакетів, що мають адресу джерела з підмережі 127.0.0.0/8 і не з петлевого (loopback) інтерфейсу.

Для нейтралізації загрози У3 в загальному випадку потрібно застосування додаткових програм, які б визначали факти невдалих спроб авторизації, визначали IP адреси атакуючих, а потім за допомогою брандмауера блокували б їх по IP. Реалізації захисту від загрози У2.1. частково ускладнить зловмисникам можливість реалізації загрози У3.

Боротьба з У4 проводиться шляхом обмеження вихідного мережного трафіку, що може здійснюватися на підставі аналізу:

1. Адрес призначення (білі або чорні списки).
2. Протоколів транспортного рівня (TCP/UDP) та номерів їх портів (білі чи чорні списки).
3. Додатків, які намагаються встановити з'єднання.

Переходячи від ідей щодо захисту до їх реалізації, слід пам'ятати базову аксіому: чим надійніший захист, тим дорожчий він коштує. Причому ця вартість виражається у витратах на купівлю засобів захисту, а й трудовитратах з їхньої експлуатацію особливо із боку пересічних користувачів. Якщо захист тероризує користувача регулярними запитами або іншим чином відволікає від роботи, він усіма правдами і неправдами саботуватиме її застосування (user resistance). Тому завжди потрібно знаходити баланс між захищеністю та зручністю. Для цього в деяких випадках слід приймати (ігнорувати) ризики малоймовірних або незначних по шкоді загроз інформаційній безпеці.

3.3 Схеми розмежування трафіку за рівнем захищеності та простоти реалізації

Схема 1. Мінамальний захист робочої станції

1. Заборонено весь вхідний трафік, крім того, що стосується вже встановлених з'єднань.
2. Весь вихідний трафік дозволено.

Даний схема підходить тільки для робочих станцій з мінімальними очікуваннями з безпеки. У таких робочих станціях немає активних мережевих сервісів (не повинно бути), а брандмауер захищає від несанкціонованого відкриття мережевих портів шкідливими. Непрямо брандмауер також реалізує «захист від дурня», що полягає в тому, що якщо користувач з необережності або в тестових цілях встановить будь-який мережевий сервіс, то по мережі він буде недоступний.

Обмеження вихідного трафіку за адресами призначення на робочих станціях, де активно користуються Інтернетом, є вкрай трудомістким, а обмежувати трафік за додатками брандмауер nftables не вміє. Відповідно весь вихідний мережевий трафік дозволяється без обмежень. Загроза У3 повністю ігнорується. Як слабкий варіант боротьби з У3 можна рекомендувати застосування вбудованої системи мандатного розмежування доступу (МАС) AppArmor, яка дозволяє вибраним програмам вимкнути мережні можливості. Проблема в тому, що AppArmor працює тільки за чорними списками, білий список - замкнуте програмне середовище - з його допомогою не вийде.

Розглянута схема розмежування трафіку дозволяє повністю нейтралізувати небезпеки У1, У2, У3, У5, небезпека У4 ігнорується. За промовчанням вбудований брандмауер Windows працює за цією схемою.

Схема 2. Мінімальний захист сервера.

Схема практично повністю повторює попередню, за винятком того, що:

1. Дозволяється вхідний трафік по вибраним мережевим портам, необхідним роботи мережевих сервісів.
2. Блокується вхідний трафік для IP-пакетів, адреса джерела яких відноситься до мережі 127.0.0.0/8, і які не прийшли з петлевого (loopback) інтерфейсу.

Обмеження доступу до мережевих сервісів за IP-адресами не здійснюється. Реалізація даної схеми частково захищає від У1 і повністю від У2.2 і У5, інші загрози ігноруються.

Схема 3. Стандартний захист серверів

Обмеження вхідного трафіку у цій схемі повністю повторює попередню схему. Вихідний трафік повністю забороняється, за винятком трафіку:

1. Поточного за вже встановленими з'єднаннями;
2. Що надсилається за переліком явно дозволених портів протоколів транспортного рівня (білий список).

Подібна схема фільтрації повністю захищає від У2.2 та У5, частково від інших загроз, крім У2.1, яка повністю ігнорується.

Схема 4. Посилений захист серверів

Схема заснована на попередній, але посилена такими заходами:

1. Для кожного мережевого сервісу (за наявності можливості) налаштовується:
    a. Обмеження максимальної частоти (rate) спроб підключення.
    b. Обмеження щодо переліку вузлів, які мають право на підключення (білий список).

c. Використання додаткових засобів виявлення великої кількості невдалих спроб авторизації, після чого порушника за допомогою брандмауера блокують IP.

2. Вихідний трафік, окрім портів, обмежується також і за IP-адресами призначення.

Подібна схема фільтрації дає максимальний захист від розглянутих загроз.

3.4 Створення лабораторного стенду корпоративної інформаційної системи

Давайте тепер зберемо лабораторний стенд (Рис. 3.2) та відпрацюємо на ньому наші ідеї. Тут ми навіть трохи ускладнимо завдання. У FW буде не два інтерфейси: один внутрішній і один зовнішній, а чотири: один зовнішній і три внутрішні. Всі вузли, підключені до внутрішніх інтерфейсів, вважатимемо внутрішнім сегментом і фільтрувати трафік між цими вузлами не будемо. Як ви побачите далі, кількість внутрішніх інтерфейсів не має значення, але зовнішній може бути лише один.

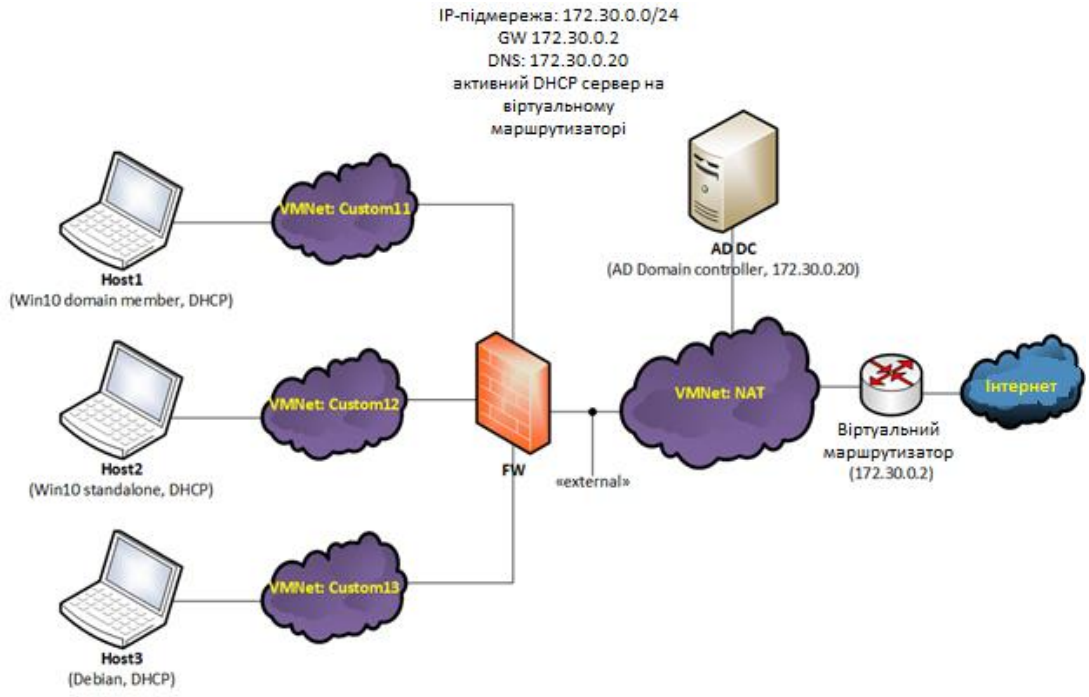

Рис. 3.2 Лабораторний стенд

У цьому прикладі для наочності ми моделюємо класичну мережу на базі ОС Windows. Host1 – це Windows 10, підключений до домену Active Directory, контролер якого (ADDC) розташований у зовнішній мережі. Host2 – Windows 10, не підключена до домену, а Host3 – наша шаблонна машина на базі Debian. Всі HostX отримують IP-адреси DHCP від віртуального маршрутизатора.

Для моделювання наших ідей за допомогою засобів віртуалізації, кожен вузол внутрішньої мережі підключимо через окрему віртуальну мережу до FW. Як і минулого разу, ми робимо це для того, щоб весь трафік між вузлами HostX і вузлами зовнішньої мережі йшов через FW.

Захистемо вузли HosX від несанкціонованих підключень:

1. На вузлі FW встановимо 4 мережеві інтерфейси. Для наочності за допомогою засобів віртуалізації змінимо на них MAC-адреси:

```
root@debian:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: external: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast master br0 state UP group d
efault qlen 1000
    link/ether 00:11:11:11:11:00 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
3: ens36: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast master br0 state UP group defa
ult qlen 1000
    link/ether 00:22:22:22:22:00 brd ff:ff:ff:ff:ff:ff
    altname enp2s4
4: ens37: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast master br0 state UP group defa
ult qlen 1000
    link/ether 00:33:33:33:33:00 brd ff:ff:ff:ff:ff:ff
    altname enp2s5
5: ens38: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast master br0 state UP group defa
ult qlen 1000
    link/ether 00:44:44:44:44:00 brd ff:ff:ff:ff:ff:ff
    altname enp2s6
6: br0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether 3a:ca:3c:75:76:c1 brd ff:ff:ff:ff:ff:ff
    inet6 fe80::38ca:3cff:fe75:76c1/64 scope link
       valid_lft forever preferred_lft forever
root@debian:~#
```

Рис. 3.4 Мережеві інтерфейси

2. Один з інтерфейсів з MAC-адресою 00:11:11:11:11:00, підключений до NAT, перейменуємо в external. Для цього створимо файл /etc/systemd/network/10-set-external-name.link та заповнимо його наступним змістом:

```
# /etc/systemd/network/10-set-internal1-name.link
[Match]
MACAddress=00:11:11:11:11:00

[Link]
Name=external
```

Рис. 3.5 Вміст link файлу

3. Як і при вирішенні попереднього завдання встановимо на вузол FW пакет bridge-utils.
4. Зробимо із FW комутатор. Для цього у файл /etc/network/interfaces запишемо наступний зміст:

```
<table><tr><td>
auto lo
iface lo inet loopback

iface external inet manual

iface ens36 inet manual

iface ens37 inet manual

iface ens38 inet manual

auto br0
iface br0 inet manual
bridge_ports external ens36 ens37 ens38
```

Рис. 3.6 Модифікація файлу interfaces

5. Перевіримо працездатність мережі, “пінгуючи” всі вузли. Для проведення тестування не забудьте вимкнути вбудований міжмережевий екран на машинах Windows. Усі вузли повинні “пінгуватися”.

3.5 Використання Nftables правил

Nftables - дуже потужний інструмент, що функціоналом істотно перевершує міжмережевий екран у класичному його розумінні. Як завдання на саморозвиток пропоную розробити конфігурацію локального міжмережевого екрану, яка виявляла б факти сканування портів і блокувала порушників по IP на 15 хвилин. Про отримані результати напишіть у коментарях.

У Red Hat Enterprise Linux 8 пріоритетним низькорівневим рішенням є nftables. У цій статті ми поговоримо про те, як почати використовувати nftables. Найбільш актуальною вона буде для системних адміністраторів та DevOps-фахівців. Там, де це має сенс, ми поговоримо про різницю між nftables і його попередником — iptables.

Для початку необхідно відзначити, що nftables – це userland-утиліта, nft та підсистема ядра. Усередині ядра вона будується з урахуванням підсистеми netfilter. У цій статті ми говоритимемо про взаємодію користувача з утилітою nft.
Отже, як виглядатиме налаштування nftables за замовчуванням? Давайте з'ясуємо, вивівши весь набір правил за допомогою команди nft list ruleset. В результаті ми отримаємо нічого.

За умовчанням nftables не створює таблиці та ланцюжка, на відміну від свого попередника iptables. Порожній набір правил має нульову вартість ресурсів. Для порівняння, згадайте iptables, де кожна попередньо створена таблиця та ланцюжок мали враховуватися, а базові лічильники пакетів збільшувалися, навіть якщо в них було порожньо.

У nftables нам доведеться створювати таблиці вручну. Таблиці повинні визначати сімейство: IP, IP6, INET, ARP, BRIDGE чи NETDEV. Тут INET означає, що таблиця оброблятиме пакети ipv4 і ipv6. Саме цю родину ми й використовуватимемо у статті.

У nftables вам доведеться створювати таблиці вручну. Таблиці повинні визначати сімейство: ip, ip6, inet, arp, bridge чи netdev. Тут inet означає, що таблиця оброблятиме пакети ipv4 і ipv6. Саме цю родину ми й використовуватимемо у

статті. Давайте створимо нашу першу таблицю та перерахуємо набір правил (Рис. 3.7)

```
# nft add table inet my_table
# nft list ruleset
table inet my_table {
}
```

Рис. 3.7 Набір правил

Відмінно тепер у нас є таблиця, але сама по собі вона мало що дає. Давайте перейдемо до ланцюжків. Ланцюжки - об'єкти, які будуть містити правила брандмауера.

За аналогією з таблицями, ланцюжки також потрібно створювати вручну. При створенні ланцюжка необхідно вказати, до якої таблиці він відноситься, а також тип, хук та пріоритет. У цьому посібнику ми нічого не ускладнюватимемо і використовуємо filter, input, і priority 0 для фільтрації пакетів, призначених для хоста.

```
nft add chain inet my_table my_filter_chain { type filter hook input priority 0 \; }
```

Рис. 3.8 Створення ланцюжка

Зворотний слеш () потрібен, щоб shell не інтерпретував дужку як кінець команди. Ланцюжки також можуть бути створені без вказівки хука. Створені таким чином ланцюжки еквіваленти ланцюжкам, створеним користувачами iptables. Правила можуть використовувати оператори jump або goto для виконання правил у ланцюжку. Ця механіка корисна для логічного поділу правил або для того, щоб

ділитися підмножиною правил, які інакше продублювалися б. Можливо це зробити за допомогою команди “nft add chain inet my_table my_utility_chain”.

Тепер, коли ви створили таблицю та ланцюжок, ви можете нарешті додати правила для брандмауера. Давайте додамо правило для дозволу SSH. Використовуємо правило “nft add rule inet my_table my_filter_chain tcp dport ssh accept”.

Тут слід зазначити, що як ми додали його в таблицю сімейства inet, це єдине правило оброблятиме як пакети IPv4, так і пакети IPv6. Дієслово add додаватиме правило в кінець ланцюжка. Також можна використовувати дієслово insert, щоб додати правило на початок ланцюжка.

Ми додали два правила, а тепер давайте подивимося, як виглядатиме повний набір правил.

```
# nft list ruleset
table inet my_table {
chain my_filter_chain {
type filter hook input priority 0; policy accept;
tcp dport http accept
tcp dport ssh accept
}
}
```

Рис. 3.9 Додані правила в nft

Зверніть увагу, що правило http має відпрацьовувати раніше правила ssh, оскільки ми використовували вище дієслово insert. Також ми можемо додати правило довільне місце в ланцюжку. Є два способи зробити це:

1. Використовуйте index, щоб вказати на індекс у списку правил. Використання add додасть нове правило після зазначеного індексу. Якщо ви використовуєте insert, то нове правило буде додано перед правилом із заданим індексом. Значення індексів розпочинаються з 0.

```
# nft insert rule inet my_table my_filter_chain index 1 tcp dport nfs accept
# nft list ruleset
table inet my_table {
chain my_filter_chain {
type filter hook input priority 0; policy accept;
tcp dport http accept
tcp dport nfs accept
tcp dport ssh accept
}
}

# nft add rule inet my_table my_filter_chain index 0 tcp dport 1234 accept
# nft list ruleset
table inet my_table {
chain my_filter_chain {
type filter hook input priority 0; policy accept;
tcp dport http accept
tcp dport 1234 accept
tcp dport nfs accept
tcp dport ssh accept
}
}
```

Рис. 3.10 Використання index

Використання index з insert за фактом еквівалентно опції iptables -I з індексом. Перше, що потрібно пам'ятати, так це про те, що індекси в nftables починаються з 0. По-друге, індекс повинен вказувати на існуюче правило. Тобто писати "nft insert rule … index 0" у порожньому ланцюжку неприпустимо.

2. Використовуйте handle, щоб вказати правило, до чи після якого потрібно вставляти інше правило. Для вставки після використання дієслова add. Щоб вставити до правила, використовуйте insert. Ви можете отримати handle правил, додавши прапор Handle при виведенні правил.

```
# nft --handle list ruleset
table inet my_table { # handle 21
chain my_filter_chain { # handle 1
type filter hook input priority 0; policy accept;
tcp dport http accept # handle 3
tcp dport ssh accept # handle 2
}
}
# nft add rule inet my_table my_filter_chain handle 3 tcp dport 1234 accept
# nft insert rule inet my_table my_filter_chain handle 2 tcp dport nfs accept
# nft --handle list ruleset
table inet my_table { # handle 21
chain my_filter_chain { # handle 1
type filter hook input priority 0; policy accept;
tcp dport http accept # handle 3
tcp dport 1234 accept # handle 8
tcp dport nfs accept # handle 7
tcp dport ssh accept # handle 2
}
}
```

Рис. 3.11 Використання handle

У nftables handle правил стабільний і не зміниться доти, доки правило не буде видалено. Так посилання на правило виходить надійніше, ніж просто індекс, який може змінитися при вставці іншого правила. Також ви можете отримати handle правила під час створення, використовуючи відразу два прапори –echo та –handle. Правило буде виведене у CLI разом з його handle.

```
# nft --echo --handle add rule inet my_table my_filter_chain udp dport 3333 accept
add rule inet my_table my_filter_chain udp dport 3333 accept # handle 4
```

Рис. 3.12 Додавання правила

Старі версії nftables використали позицію ключового слова. З того часу механіка ключових слів застаріла та з'явився handle.

Видалення правил виконується за допомогою handle правила за аналогією з командами add та insert вище. Спочатку потрібно знайти handle правила, які ви хочете видалити. Потім використовуйте цей handle для видалення правила.

У nftables є нативна підтримка наборів. Вони можуть бути корисними, якщо ви хочете, щоб правило працювало з кількома IP-адресами, портами, інтерфейсами або за будь-якими іншими критеріями. Будь-яке правило може мати inline-набори. Ця механіка корисна для наборів, які ви не збираєтесь змінювати. Наприклад, наступна команда буде дозволяти весь трафік з 10.10.10.123 та 10.10.10.231.

```
# nft add rule inet my_table my_filter_chain ip saddr
# nft list ruleset
table inet my_table {
chain my_filter_chain {
type filter hook input priority 0; policy accept;
tcp dport http accept
tcp dport nfs accept
tcp dport ssh accept
ip saddr { 10.10.10.123, 10.10.10.231 } accept
}
```

Рис. 3.13 Дозвіл трафіку з діапазону IP адрес

Недоліком цього є те, що якщо вам потрібно змінити набір, то потрібно буде замінювати і правило. Щоб отримати мутабельні набори потрібно використовувати іменовані набори. Як інший приклад, замість перших трьох правил ми могли б використовувати анонімний набір.

```
# nft add rule inet my_table my_filter_chain tcp dport { http, nfs, ssh } accept
```

Рис. 3.14 Анонімний набір

Користувачі iptables швидше за все звикли до використання ipset. Оскільки nftables має власну нативну підтримку наборів, ipset використовувати не потрібно. Nftables також підтримує мутабельні іменовані набори. Для їх створення необхідно вказати тип елементів, які у них містяться. Наприклад, типи можуть бути такими: ipv4_addr, inet_service, ether_addr. Давайте створимо порожній набір.

```
# nft add set inet my_table my_set {type ipv4_addr\; }
# nft list sets
table inet my_table {
set my_set {
type ipv4_addr
}
}
```

Рис. 3.15 Порожній набір

Щоб послатися на набір у правилі, використовуємо символ @ та ім'я набору після нього. Наступне правило буде працювати як чорний список для IP-адрес у нашому наборі.

```
# nft insert rule inet my_table my_filter_chain ip saddr @my_set drop
# nft list chain inet my_table my_filter_chain
table inet my_table {
chain my_filter_chain {
type filter hook input priority 0; policy accept;
ip saddr @my_set drop
tcp dport http accept
tcp dport nfs accept
tcp dport ssh accept
ip saddr { 10.10.10.123, 10.10.10.231 } accept
}
}
```

Рис. 3.16 Чорний список IP-адрес

Звісно, це особливо ефективно, оскільки у наборі порожньо. Давайте додамо до нього кілька елементів.

```
# nft add element inet my_table my_set { 10.10.10.22, 10.10.10.33 }
# nft list set inet my_table my_set
table inet my_table {
set my_set {
type ipv4_addr
elements = {10.10.10.22, 10.10.10.33}
}
}
```

Рис. 3.17 Додаємо кілька правил в набір

Однак, спроба додати діапазон значень приведе до помилки. Щоб використовувати діапазони в наборах, потрібно створити набір з використанням інтервалів прапорів. Так потрібно, щоб ядро заздалегідь знало, який тип даних зберігатиметься в наборі, щоб використовувати відповідну структуру даних.

У наборах також можна використовувати діапазони. Для IP-адрес це може бути вкрай корисно. Для використання діапазонів набір має бути створений з використанням прапорів інтервалів.

```
# nft add set inet my_table my_range_set {type ipv4_addr\; flags interval \; }
# nft add element inet my_table my_range_set { 10.20.20.0/24 }
# nft list set inet my_table my_range_set
table inet my_table {
set my_range_set {
type ipv4_addr
flags interval
elements = {10.20.20.0/24}
}
}
```

Рис. 3.18 Використання діапазонів IP-адрес

Нотація маски мережі була неявно перетворена на діапазон IP-адрес. Аналогічно ми могли б написати 10.20.20.0-10.20.20.255 і отримати той же ефект. Набори також підтримують агрегатні типи та збіги. Тобто елемент набору може містити кілька типів, а правило може використовувати оператор конкатенації «.» при зверненні до набору. У цьому прикладі ми зможемо зіставляти IPv4-адреси, IP-протоколи та номери портів одночасно.

```
# nft add set inet my_table my_concat_set {type ipv4_addr. inet_proto. inet_service\; }
# nft list set inet my_table my_concat_set
table inet my_table {
set my_concat_set {
type ipv4_addr. inet_proto. inet_service
}
}
```

Рис. 3.19 Зіставлення портів та протоколів

Тепер можна додати елементи до списку “nft add element inet my_table my_concat_set { 10.31.31.31 . tcp. telnet }”. Як бачимо, символьні імена (tcp, telnet) також можна використовувати при додаванні елементів набору. Використання набору в правилі аналогічно до іменованого набору вище, але правило повинно виконувати конкатенацію.

```
# nft add rule inet my_table my_filter_chain ip saddr. meta l4proto. tcp dport @my_concat_set accept
# nft list chain inet my_table my_filter_chain
table inet my_table {
chain my_filter_chain {
...
ip saddr { 10.10.10.123, 10.10.10.231 } accept
meta nfproto ipv4 ip saddr. meta l4proto. tcp dport @my_concat_set accept
}
}
```

Рис. 3.20 Конкатинація правил

Також варто зазначити, що конкатенація може використовуватись з inline-наборами. Ось останній приклад, що відбиває це. Конкатенації наборів nftables аналогічні агрегатним типам ipset, наприклад, hash: ip, port.

ВИСНОВКИ

Підсумовуючи, можна відзначити, що системи безпеки роблять значні кроки вперед, стаючи необхідним IT-інструментом сучасних підприємств. Пов'язано це з багатьма чинниками. Репутаційні ризики та ризики відмови в обслуговуванні клієнтів багаторазово зросли. Люди все частіше роблять покупки через Інтернет, отримують послуги/держпослуги та ін. Компанії більше не можуть дозволити собі децентралізовані системи, коли над одним і тим же завданням у різних регіонах, у різних філіях компанії працює велика кількість співробітників. Відбувається централізація ресурсів, все більше компаній починають використовувати хмарні послуги та послуги, що сильно змінює бізнес-процеси, а відповідно, і IT-інфраструктуру компаній.

Для забезпечення інформаційної безпеки для корпоративного сектора перспективною технологією є NGFW, за допомогою якої забезпечується детальний контроль, що налаштовується на рівні додатків. Технологія міжмережевих екранів була значно вдосконалена - вона еволюціонувала до спеціалізованих рішень, що виконують глибокий аналіз трафіку та ідентифікацію програм. Відповідні продукти почали працювати швидше і підтримують складніші набори правил, ніж їхніх попередників.

Як показує практика, традиційні міжмережеві екрани досі потрібні. В основному це пов'язано з обмеженим контролем за реалізацією сервісів безпеки з боку регулюючих органів та із забезпеченням захисту IT за залишковим принципом - дешевше та за мінімумом. Тому місце для традиційних екранів, безумовно, є, проте їх оснащуватимуть новими функціями. Наприклад, більш затребуваними стають пристрої, в яких, крім традиційного FW, є ще й засоби поглибленого аналізу міжмережевих потоків.

Однак міжмережеві екрани – не панацея. При виборі рішення потрібно чітко визначити, які саме функції необхідні, переконатися в тому, що заявлені вендором захисні властивості можуть бути реалізовані в конкретному робочому середовищі, що в компанії достатньо ресурсів для управління політиками безпеки.

Незважаючи на основну проблему мереж VPN, пов'язану з відсутністю усталених стандартів аутентифікації та обміну шифрованою інформацією, ця технологія досі затребувана та обладнання VPN користується активним попитом на ринку засобів інформаційного захисту.

Ще більш затребуваним є обладнання для запобігання DDOS-атакам - це пов'язано зі значним збільшенням шкідливих ботнетів по всьому світу за останні 3 роки.

За статистикою, щонайменше 75% усіх комп'ютерних злочинів відбувається всередині корпоративної мережі – з вини співробітників підприємства. Традиційні засоби захисту (наприклад, міжмережеві екрани FW) не дозволяють захистити корпоративну мережу від наміру зловмисника, який має у ній доступ у рамках робочих повноважень. Для побудови ефективної системи захисту інформації потрібні додаткові засоби запобігання атакам.

Такими засобами стали програмно-апаратні комплекси із застосуванням технологій: IPS (IDS), NAC, IDM, VPN. Необхідно також використовувати сучасні засоби автентифікації та авторизації.

Хоча IPS/IDS і міжмережевий екран відносяться до засобів забезпечення інформаційної безпеки, міжмережевий екран відрізняється тим, що обмежує надходження на хост або підмережу певних видів трафіку для запобігання вторгненням і не відстежує вторгнення, що відбуваються всередині мережі. IPS/IDS, навпаки, пропускає трафік, аналізуючи його та сигналізуючи при виявленні підозрілої активності. Виявлення порушення безпеки проводиться зазвичай з використанням евристичних правил та аналізу сигнатур відомих комп'ютерних атак.

Використання IDM-рішень має на увазі наявність повномасштабної корпоративної рольової моделі користувачів. У ній враховуються всі інформаційні активи підприємства, а також описуються бізнес-ролі персоналу та порядок доступу для кожної з них до кожного активу. При цьому узгодження моделі, як правило, пов'язане з великими труднощами через суперечливі вимоги до неї з боку різних структур та підрозділів компанії.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Nftables — Вікіпедія [Електронний ресурс] – Режим доступу: https://uk.wikipedia.org/wiki/Nftables
2. nftables wiki [Електронний ресурс] – Режим доступу: https://wiki.nftables.org/wiki-nftables/index.php/Main_Page
3. nftables – ArchWiki [Електронний ресурс] – Режим доступу: https://wiki.archlinux.org/title/Nftables
4. netfilter/iptables project homepage [Електронний ресурс] – Режим доступу: https://netfilter.org/projects/nftables/
5. nftables – записки адміна [Електронний ресурс] – Режим доступу: https://sysadmin.pm/nftables/
6. nftables – Debian Wiki [Електронний ресурс] – Режим доступу: https://wiki.debian.org/nftables
7. [OpenWrt Wiki] nftables [Електронний ресурс] – Режим доступу: https://openwrt.org/docs/guide-user/firewall/misc/nftables
8. Міжмережевий екран - захист локальної мережі [Електронний ресурс] – Режим доступу: https://sites.google.com/site/zahistlokalnoiemerezi/zahist/mizmerezevij-ekran
9. Міжмережевий екран: що це таке і для чого він потрібен [Електронний ресурс] – Режим доступу: https://texnogid.biz.ua/wi-fi/bezpeka/mizhmerezhevyj-ekran.html
10. Як працює міжмережевий екран? – Keenetic [Електронний ресурс] – Режим доступу: https://help.keenetic.com/hc/uk/articles/360001429839
11. Linux Firewalls: Enhancing Security with nftables and Beyond: Enhancing Security with nftables and Beyond (4th Edition) Steve Suehring, 2015. 201 с.
12. Security - Firewall | Ubuntu [Електронний ресурс] – Режим доступу: https://ubuntu.com/server/docs/security-firewall
13. Top 10 Linux Firewall Solutions in 2021 | Spiceworks 1 [Електронний ресурс] – Режим доступу: https://www.spiceworks.com/it-security/network-security/articles/top-10-linux-firewall-solutions/

14. Настройка Firewall [Електронний ресурс] – Режим доступу: https://docstore.mik.ua/manuals/ru/install_redhat/s1-firewallconfig.html

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (ПРЕЗЕНТАЦІЯ)