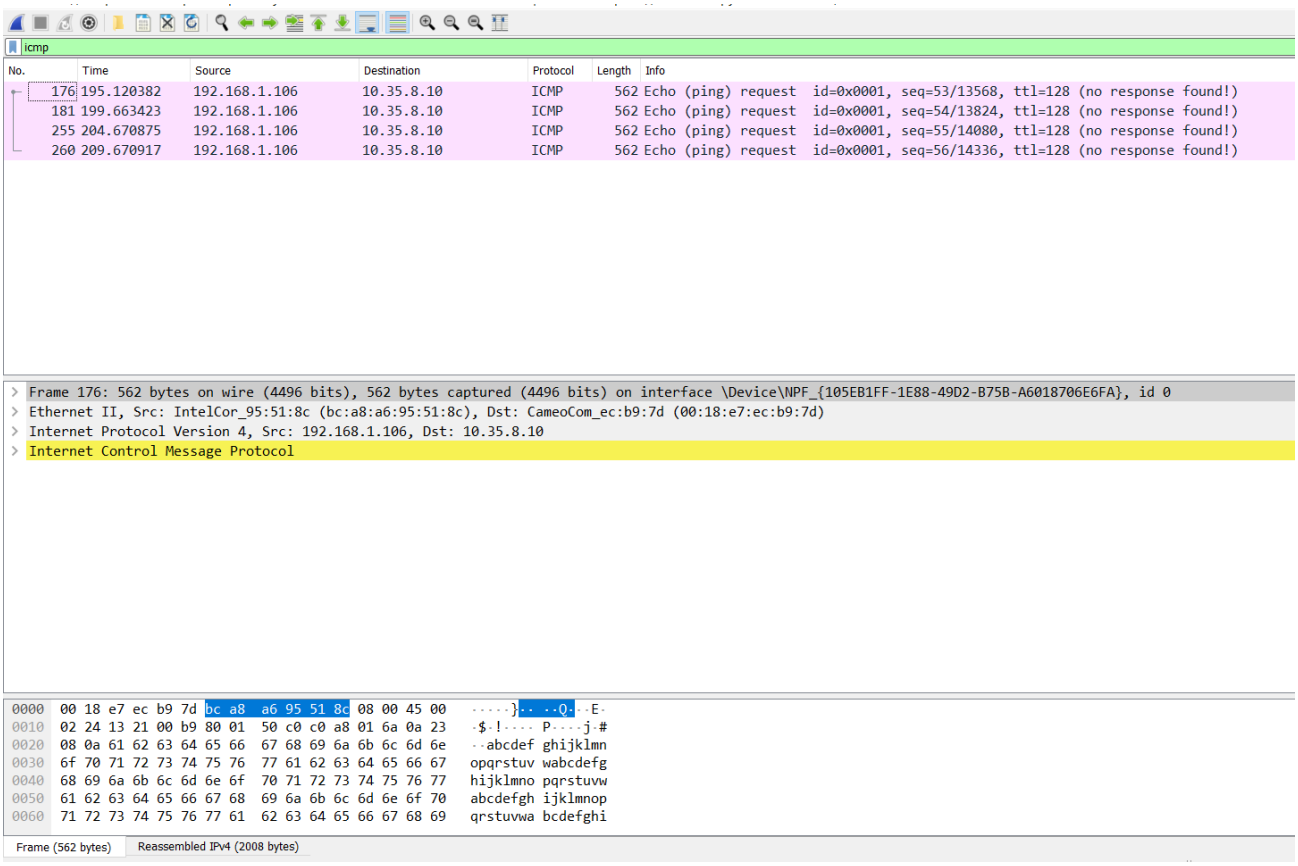
«ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ»
НАЦІОНАЛЬНОГО ТЕХНІЧНОГО УНІВЕРСИТЕТУ УКРАЇНИ «КПІ»
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

Лабораторна робота №5

з курсу «Комп'ютерні мережі»

Виконала: студентка 3 курсу
групи КА 73
Клименко А.І.
Прийняв: Кухарєв С.О.

Київ 2020 р.



(Термінал:

Значок Windows + R -> Cmd)

Frame 176: 562 bytes on wire (4496 bits), 562 bytes captured (4496 bits) on interface \Device\NPF_{105EB1FF-1E88-49D2-B75B-A6018706E6FA}, id 0

Ethernet II, Src: IntelCor_95:51:8c (bc:a8:a6:95:51:8c), Dst: CameoCom_ec:b9:7d (00:18:e7:ec:b9:7d)

Internet Protocol Version 4, Src: 192.168.1.106, Dst: 10.35.8.10

0100 .... = Version: 4

.... 0101 = Header Length: 20 bytes (5)

Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)

Total Length: 548

Identification: 0x1321 (4897)

Flags: 0x00b9

0... .... = Reserved bit: Not set

.0.. .... = Don't fragment: Not set

..0. .... = More fragments: Not set

...0 0101 1100 1000 = Fragment offset: 1480

Time to live: 128

Protocol: ICMP (1)

Header checksum: 0x50c0 [validation disabled]

[Header checksum status: Unverified]

Source: 192.168.1.106

Destination: 10.35.8.10

[2 IPv4 Fragments (2008 bytes): #175(1480), #176(528)]

[Frame: 175, payload: 0-1479 (1480 bytes)]

[Frame: 176, payload: 1480-2007 (528 bytes)]

[Fragment count: 2]

[Reassembled IPv4 length: 2008]

[Reassembled IPv4 data: 08007b4200010035616263646565666768696a6b6c6d6e6f70…]

Internet Control Message Protocol

Контрольні запитання:

1. Визначте IP адреси вашої та цільової робочих станцій.

192.168.1.106,

10.35.8.10 .

2. Яке значення в полі номера протоколу вищого рівня в заголовку IP першого пакету із запитом ICMP?

Protocol: ICMP (1)

3. Скільки байт займає заголовок IP першого пакету із запитом ICMP? Скільки байт займає корисна інформація (payload) пакету? Поясніть як ви встановили кількість байт корисної інформації.

20 байт, 2008 байт = 1480 + 528 ;

4. Дослідіть пакет із пунктів 2/3. Чи фрагментований цей пакет? Поясніть як ви встановили фрагментацію пакету. Як можна встановити номер фрагменту, що передається у пакеті?

Так, фрагментований. Встановлено за номером фрейма.

[2 IPv4 Fragments (2008 bytes): #175(1480), #176(528)]

5. Знайдіть наступний фрагмент датаграми IP. Яка інформація дозволяє встановити наявність наступних фрагментів, що мають слідувати за другим фрагментом?

More fragments: Not set

6. Які поля протоколу IP відрізняють перший фрагмент від другого?

Відрізняють назва фрейму; Upper Layer Protocol, Fragment offset.

7. Розгляньте послідовність пакетів IP із запитами ICMP вашої робочої станції. Які поля заголовку IP завжди змінюються?

Identification та Header checksum ;

8. Розгляньте послідовність пакетів IP із запитами ICMP вашої робочої станції. Які поля заголовку IP мають зберігати свої значення? Які поля мають змінюватися?

Поля, які зберігають свої значення*:*
1) Version (ми використовуємо IPv4 для всіх пакетів)
2) header length (всі пакети – ICMP)
3)source IP, destination IP (Ми пінгуємо одну і ту ж адресу)
4) Differentiated Services (всі ICMP пакети одного службового типу)
5) Time to live

Поля, які змінюють свої значення:
6) Upper Layer Protocol (всі загаловки ICMPмають унікальні поля, що змінюються)
7) Identification (IP пакети мають мати різні id)
8) Header checksum (оскільки заголовки змінюються, то контрольна сума

9. Розгляньте послідовність пакетів IP із запитами ICMP вашої робочої станції. Опишіть закономірність зміни значень поля Identification рівня IP.

Значення кожен раз збільшується на 1 ;

10. Розгляньте послідовність пакетів IP із повідомленнями TTL-exceeded від найближчого маршрутизатора. Які значення встановлені у полях Identification та TTL?

Даних послідовностей пакетів не було. (Утиліта ping не змінює TTL для різних запитів) ;

11. Розгляньте послідовність пакетів IP із повідомленнями TTL-exceeded від найближчого маршрутизатора. Які значення встановлені у полях Identification та TTL? Чи змінюються ці значення для різних пакетів у послідовності?

Для кожної ICMP TTL-exceeded відповіді змінюється поле Identification. Якщо дві IP датаграми мають однакове поле Identification, то дані датаграми є фрагментами однієї великої IP датаграми. Поле TTL завжди мусить бути однакове, адже у заданого маршрутизатора він один.

Утиліта IP tables

Iptables - утиліта командного рядка, є стандартним інтерфейсом управління роботою брандмауера (Брандмауер — програма чи пристрій, що здійснює захист комп'ютерних мереж)

Netfilter для ядер Linux, починаючи з версії 2.4.

З її допомогою адміністратори створюють і змінюють правила, що керують фільтрацією і перенаправленням пакетів.

Ключовими поняттями iptables є:

Правило - складається з критерію, дії і лічильника. Якщо пакет відповідає критерію, до нього застосовується дія, і він враховується лічильником. Критерію може і не бути - тоді неявно передбачається критерій «всі пакети». Вказувати дію теж не обов'язково - за відсутності дії правило буде працювати тільки як лічильник. Правила для кожного ланцюжка спрацьовують в порядку їх слідування, тому порядок важливий.

Критерій - логічне вираження, що аналізує властивості пакета і / або з'єднання і визначає , чи підпадає даний конкретний пакет під дію поточного правила.

Дія - опис дії, яку треба виконати з пакетом і / або з'єднанням в тому випадку, якщо вони підпадають під дію цього правила.

Лічильник - компонент правила, що забезпечує облік кількості пакетів, які потрапили під критерій даного правила. Також лічильник враховує сумарний обсяг таких пакетів в байтах.

Ланцюжок - упорядкована послідовність правил. Ланцюжки можна розділити на призначені для користувача і базові.

Базовий ланцюжок - ланцюжок, що створюється за замовчуванням при ініціалізації таблиці. Крім того, базовий ланцюжок відрізняється від призначеного для користувача наявністю «дії за замовчуванням».

// Кожен пакет, в залежності від того, чи призначений він самому хосту, //згенерований ним або є транзитними, повинен пройти покладений йому //набір базових ланцюжків різних таблиць.

// Ця дія застосовується до тих пакетів, що не були оброблені іншими правилами цього // ланцюжка і викликаних з неї ланцюжків.

Імена базових ланцюжків завжди записуються в верхньому регістрі

/// (PREROUTING, INPUT, FORWARD, OUTPUT, POSTROUTING).

Ланцюжок користувача - ланцюжок, створений користувачем. Може використовуватися тільки в межах своєї таблиці.

// Рекомендується не використовувати для таких ланцюжків імена у
// верхньому регістрі, щоб уникнути плутанини з базовими ланцюжками і
// вбудованими діями.

Таблиця - сукупність базових і призначених для користувача ланцюжків, об'єднаних загальним функціональним призначенням. Імена таблиць записуються в нижньому регістрі, так як в принципі не можуть конфліктувати з іменами призначених для користувача ланцюжків. При виклику команди iptables таблиця вказується в форматі -t ім'я_таблиці. При відсутності явної вказівки, використовується таблиця filter.

Архітектура:

В системі netfilter, пакети пропускаються через ланцюжки. Ланцюжок є впорядкованим списком правил, а кожне правило може містити критерії і дію або перехід. Коли пакет проходить через ланцюжок, система netfilter по черзі перевіряє, чи відповідає пакет всіма критеріями чергового правила, і якщо так, то виконує дію (якщо критеріїв в правилі немає, то дія виконується для всіх пакетів проходять через правило). Варіантів можливих критеріїв дуже багато. Наприклад, пакет відповідає критерію -source 192.168.1.1 якщо в заголовку пакета вказано, що відправник - 192.168.1.1. Найпростіший тип переходу, -jump, просто пересилає пакет в початок іншої ланцюжка. Також за допомогою -jump можна вказати дію. Стандартні дії доступні у всіх ланцюжках - ACCEPT (пропустити), DROP (видалити), QUEUE (передати на аналіз зовнішньої програмі), і RETURN (повернути на аналіз в попередню ланцюжок). Наприклад, команди

```
iptables -A INPUT --source 192.168.1.1 --jump ACCEPT
iptables -A INPUT --jump other_chain
```

означають «додати до кінця ланцюжка INPUT наступні правила: пропустити пакети з 192.168.1.1, а все, що залишиться - відправити на аналіз в ланцюжок other_chain»

Ланцюги:

Існує 5 типів стандартних ланцюжків, вбудованих в систему:

- PREROUTING - для початкової обробки вхідних пакетів.
- INPUT - для вхідних пакетів адресованих безпосередньо локальному процесу (клієнту або сервера).
- FORWARD - для вхідних пакетів перенаправлених на вихід (зауважте, що перенаправляє пакети проходять спочатку ланцюг PREROUTING, потім FORWARD і POSTROUTING).
- OUTPUT - для пакетів генеруються локальними процесами.
- POSTROUTING - для остаточної обробки вихідних пакетів.

Також можна створювати і знищувати власні ланцюжки за допомогою утиліти iptables.

Таблицы

Ланцюжки організовані в 4 таблиці:

- raw - проглядається до передачі пакета системі визначення станів. Використовується рідко, наприклад для маркування пакетів, які НЕ повинні оброблятися системою визначення станів. Для цього в правилі вказується дію NOTRACK. Містить ланцюжка PREROUTING і OUTPUT.
- mangle - містить правила модифікації (зазвичай заголовка) IP-пакетів. Серед іншого, підтримує дії TTL (Time to live), TOS (Type of Service), і MARK (для зміни полів TTL і TOS, і для зміни маркерів пакета). Рідко необхідна і може бути небезпечна. Містить всі п'ять стандартних ланцюжків.
- nat - переглядає тільки пакети, що створюють нове з'єднання (відповідно до системи визначення станів). Підтримує дії DNAT, SNAT, MASQUERADE, REDIRECT. Містить ланцюжка PREROUTING, OUTPUT, і POSTROUTING.
- filter - основна таблиця, використовується за умовчанням якщо назва таблиці не вказано. Містить ланцюжка INPUT, FORWARD, і OUTPUT.

Ланцюжки з однаковою назвою, але в різних таблицях - абсолютно незалежні об'єкти. Наприклад, raw PREROUTING і mangle PREROUTING зазвичай містять різний набір правил; пакети спочатку проходять через ланцюжок raw PREROUTING, а потім через mangle PREROUTING.

Стани:

В системі netfilter, кожен пакет проходить через механізм визначення станів, може мати одне з чотирьох можливих станів:

- NEW - пакет відкриває новий сеанс. Класичний приклад - пакет TCP з прапором SYN.
- ESTABLISHED - пакет є частиною вже існуючого сеансу.
- RELATED - пакет відкриває новий сеанс, пов'язаний з уже відкритим сеансом. Наприклад, під час сеансу пасивного FTP, клієнт під'єднується до порту 21 сервера, сервер повідомляє клієнтові номер другого, випадково обраного порту, після чого клієнт під'єднується до другого порту для передачі файлів. У цьому випадку другий сеанс (передача файлів по другому порту) пов'язаний з уже існуючим сеансом (початкове під'єднання до порту 21).
- INVALID - всі інші пакети.

Висновок:

В лабораторній роботі я ознайомилася з деталями роботи протоколу IP, який відповідає за адресацію пакетів. Протокол IP розпізнає формат заголовка пакета (адресну частину та іншу службову інформацію включно). IP-пакети складаються з даних верхнього рівня та IP-заголовку. Під час виконання лабораторної роботи я дізналася, що вираз «Ми пінгуємо» це цілком пристойне слово і означає можливість перевірки з'єднань в мережах на основі TCP/IP.