НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра інформаційної безпеки

«До захисту допущено»
В.о. завідувача кафедри

_____ М.В.Грайворонський
(підпис)

“___”_____2019 р.

Дипломна робота

на здобуття ступеня бакалавра

з напряму підготовки 6.170101 «Безпека інформаційних і комунікаційних систем»

на тему: «Захист проміжного хоста від ампліфікації через вразливість SSDP протоколу"

Виконав: студент 4-го курсу, групи ФБ-51
(шифр групи)

Трайдакало Максим Олегович _____
(прізвище, ім’я, по батькові) (підпис)

Керівник: к.е.н., ст. викл. Ткач Володимир Миколайович _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Рецензент: _____ _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ - 2019 року

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра інформаційної безпеки

Рівень вищої освіти – перший (бакалаврський)

Напрям підготовки 6.170101 «Безпека інформаційних і комунікаційних систем»

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри
_____ М.В.Грайворонський
(підпис)
«___»_____2019 р.

ЗАВДАННЯ
на дипломну роботу студенту

Трайдакалу Максиму Олеговичу

1.Тема роботи:
«Захист проміжного хоста від ампліфікації через вразливість SSDP протоколу»

науковий керівник роботи:
к.е.н., ст. викл. Ткач Володимир Миколайович ,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «27 травня» 2019 р. № 1414-с

2. Термін подання студентом роботи 10 червня 2019 р.

3. Вихідні дані до роботи: навчальний та робочий плани.

4. Зміст роботи: - Огляд DDoS атак та UPnP протоколів.

- Стан захищеності та відомі атаки пов'язані з SSDP.

- Проведення атаки в лабораторних умовах та впровадження захисту від атаки.

Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо)
_____

6. Дата видачі завдання 21 вересня 2018р.

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів дипломної роботи	Примітка
1	Отримання завдання	21.09.18	
2	Вивчення літератури	22.09.18 - 22.12.18	
3	Написання плану роботи	22.12.18 - 30.12.18	
4	Написання першого розділу диплому	10.01.19 - 28.02.19	
5	Написання другого розділу диплому	28.02.19 – 20.04.19	
6	Проходження переддипломної практики та проведення атаки у лабораторних умовах	20.04.19 – 20.05.19	
7	Написання третього розділу диплому	20.04.19 - 20.05.19	
8	Оформлення дипломної роботи	20.05.19 – 28.05.19	
9	Предзахист дипломної роботи	28.05.19	
10	Підготовка графічної роботи	29.05.19 – 14.06.19	
11	Захист дипломної роботи	20.06.19	

Студент _____ Трайдакало М.О.
(підпис) (ініціали, прізвище)

Науковий керівник роботи _____ Ткач В М.
(підпис) (ініціали, прізвище)

РЕФЕРАТ

Обсяг роботи 61 сторінка, 18 ілюстрацій, 3 таблиці, 13 джерел літератури.

Метою роботи є створення методу захисту проміжного хоста від ампліфікації розподіленої атаки на відмову в обслуговуванні.

Об’єкт дослідження – набір UPnP протоколів.

Предмет дослідження – вразливість SSDP протоколу, яка дозволяє реалізувати DDoS атаку.

Під час розробки методу захисту були використані такі методи дослідження:

- Аналіз – для того щоб виявити особливості реалізації кожного з UPnP протоколів.
- Моделювання та експеримент – для створення Denial of Services атаки в лабораторних умовах з використанням вразливості Simple Service Discovery Protocol.
- Наукове дослідження – для перевірки теорії методу захисту проміжного хоста в ході отриманих знань під час експерименту.

Результатом роботи є готовий метод захисту проміжного хоста від амлфікації DDoS атак реалізований в Cisco Firepower NG Firewall та інших системах, які мають аналогічний функціонал. На сьогоднішній день дану вразливість можливо закрити цим методом захисту на 600000 пристроїв.

UPnP, SSDP, DDoS, підміна, IP, безпека

ABSTRACT

The work is 61 pages long, containing 18 illustrations, 3 tables and 13 sources of references.

The purpose of the researches is to create the transition host ppfgrotection from amplification using SSDP protocol.

Object of research - a set of UPnP protocols.

The subject of the study is the vulnerability of the SSDP protocol, which allows you to implement a DDoS attack.

During the development of the protection method, the following research methods were used:

- Analysis - in order to identify the peculiarities of implementing each of the UPnP protocols.
- Simulation and experiment - to create a Denial of Services attack in a laboratory environment using vulnerability Simple Service Discovery Protocol.
- Scientific research - to test the theory of the protection method of the transition host with obtained knowledge during the experiment.

The result of the work is a ready method for protecting the transition host from the amplification of DDoS attack implemented in Cisco Firepower NG Firewall and other systems that have a similar functionality. To date, this vulnerability can be closed by this method of protection for 600,000 devices.

UPnP, SSDP, DDoS, spoofing, IP, security

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

DDoS атака (англ. distributed denial-of-service) – це кібер-атака, в якій зловмисник намагається зробити певний мережний ресурс недоступним для користувачів, при цьому трафік на мережевий ресурс жертви надходить з різних джерел.

OSI модель (англ. Open Systems Interconnection) – узгоджена модель, що характеризує і стандартизує функції зв'язку телекомунікаційної або обчислювальної системи без урахування її внутрішньої структури. Її метою є сумісність різноманітних систем зв'язку зі стандартними протоколами. Модель розділяє систему зв'язку на шари абстракції. Оригінальна версія моделі має сім шарів.

UPnP (англ. Universal Plug and Play) – набір протоколів, який дає можливість користувачеві підключити пристрій до локальної мережі, і цей пристрій буде працювати, незалежно від того, чи є пристрій принтером, сканером, файловим сервером або брандмауером. Стек протоколів UPnP використовує чітко визначені Інтернет-стандарти, такі як HTTP, XML і SOAP.

SOAP (англ. Simple Object Access Protocol) - це протокол, який об'єднує XML і HTTP, щоб забезпечити механізм обміну повідомленнями та віддалених процедур. XML використовується для передачі вмісту повідомлень, тоді як HTTP використовується для передачі повідомлень до місця призначення.

SSDP (англ. Simple Service Discovery Protocol) – протокол був розроблений як просте рішення для виявлення ресурсів на основі протоколу HTTP у мережі, що не потребує конфігурації, керування або адміністрування. Якщо ресурс SSDP приймає запит виявлення HTTP Multicast over UDP (багатоадресний запит, використовуючи HTTP та UDP), який відповідає запропонованій службі, він відповідає, надіславши відповідь безпосередньо клієнту SSDP, використовуючи HTTP over UDP (одноадресна відповідь, використовуючи HTTP та UDP).

Ботнет - це велика кількість інфікованих шкідливим програмним забезпеченням комп'ютерів, підключених до мережі Інтернет, які взаємодіють один з одним і можуть контролюватися з одного місця.

ВСТУП

В сучасному світі інформаційні технології проникли майже у всі сфери життя сучасного суспільства. Невід'ємною частиною інформаційних технологій є мережа Інтернет. Мережа Інтернет дає унікальні можливості для розповсюдження та пошуку інформації, як наслідок має надзвичайно велику кількість користувачів. У зв'язку з розповсюдженням мережі Інтернет значна доля ринку надання послуг та життя людей переходить у цифровий простір. У свою чергу сервіс надання послуг має бути постійно доступним для того, щоб максимізувати свій дохід. Фактично кожна секунда роботи такого сервісу приносить кошти своєму власнику, а його недоступність навпаки – призводить до нереалізованих продаж або послуг, а також наносить шкоду у вигляді спаду репутації та зменшенні користувачів послугами у майбутньому.

Таким чином інтернет провайдери зацікавлені в безперервному та якісному доступі користувача до мережі Інтернет з метою забезпечення можливості бути конкурентним на ринку послуг. Але, на жаль, у більшості випадків провайдери не реалізують ніякого аналізу трафіку, тому що відповідні системи аналізу трафіку є занадто ресурсозатратними. Тому клієнти інтернет провайдерів здебільшого реалізують захист власноруч або за допомогою сторонніх сервісів. Проте далеко не кожен користувач освідомлений про необхідність захисту та має ресурси для реалізація захисту.

На сьогоднішній день існує досить велика кількість хмарних сервісів, які дозволяють аналізувати дані на різних рівнях моделі OSI та пом'якшувати DDoS атаки. На мою ж думку, надзвичайно важливим аспектом боротьби з розподіленими атаками на відмову в обслуговуванні є зменшення самої площі поверхні атак. Одним із напрямків реалізації розподіленої атаки на відмову в обслуговуванні є використання критичної вразливості Simple Service Discovery Protocol. Таким чином потрібно знайти метод захисту для проміжного користувача, який має вразливий пристрій. За допомогою цього методу захисту

проміжний хост буде захищений від вразливості SSDP протоколу, а, отже, захищений від того, щоб стати проміжною ланкою у розподіленій атаці на відмову в обслуговуванні.

Метою роботи є створення методу захисту проміжного хоста від ампліфікації розподіленої атаки на відмову в обслуговуванні.

Для досягнення даної мети було поставлено такі завдання:

- Проаналізувати роботу та виявити основні аспекти реалізації UPnP протоколів.
- Виявити недолік SSDP протоколу, який дає змогу реалізувати даний вид DDoS атак.
- Реалізувати дану атаку в лабораторних умовах для подальшого пошуку можливого методу захисту.
- Виявити найбільш ефективний метод захисту та довести його ефективність в ході експерименту.

Об’єкт дослідження – набір UPnP протоколів.

Предмет дослідження – вразливість SSDP протоколу, яка дозволяє реалізувати DDoS атаку.

Під час розробки методу захисту були використані такі методи дослідження:

- Аналіз – для того щоб виявити особливості реалізації кожного з UPnP протоколів.
- Моделювання та експеримент – для створення Denial of Services атаки в лабораторних умовах з використанням вразливості Simple Service Discovery Protocol.
- Наукове дослідження – для перевірки теорії методу захисту проміжного хоста в ході отриманих знань під час експерименту.

Наукова новизна роботи включає в себе вперше реалізовану систему захисту проміжного користувача від ампліфікації з використанням Cisco Firepower NG Firewall.

Практичне значення одержаних результатів – даний метод захисту готовий до використання в Cisco Firepower NG Firewall та інших системах, які мають аналогічний функціонал. На сьогоднішній день дану вразливість можливо закрити цим методом захисту на 600000 пристроїв.

1 ОГЛЯД DDOS АТАК ТА UPNP ПРОТОКОЛІВ

1.1 Огляд DDoS атак

Постійний доступ до того чи іншого веб-сайту будь-якого напрямку є одним із найважливіших факторів його розвитку. Особливо це стосується інтернет магазинів та сайтів, які продають певні послуги за кошти в режимі онлайн. Фактично кожна секунда роботи такого сервісу приносить кошти своєму власнику, а його недоступність навпаки – призводить до нереалізованих продаж або послуг, а також наносить шкоду у вигляді спаду репутації та зменшенні користувачів послугами у майбутньому.

Розподілена атака на відмову в обслуговуванні – це масштабна DDoS атака, в якій зловмисник використовує більше однієї унікальної IP-адреси. DDoS зазвичай включає більше 5 вузлів у різних мережах, менша кількість використаних вузлів у атаці може перекфаліфікувати тип атаки у DoS атаку. Стандартна атака DDoS виникає, коли зловмисники надсилають значну кількість неправильного мережевого трафіку безпосередньо до цільового сервера або мережі. Одним із способів, яким зловмисник може досягти цього, є використання ботнету для відправки трафіку. Ботнет - це велика кількість комп'ютерів-жертв, підключених до мережі Інтернет, які спілкуються один з одним і можуть контролюватися з одного місця. Коли зловмисник використовує ботнет для DDoS-атаки, вони відправляють інструкції до частини або всіх інфікованих машин, підключених до цієї бот-мережі, тим самим збільшуючи розмір атаки, а джерелом стають відразу кілька мереж і трафік надходить, можливо, з декількох країн.

Основними мотивами розподілених атака на відмову в обслуговуванні є замовлення від осіб, які переслідують особисті цілі, зокрема нанесення фінансових збитків та дискредитації сервісу надання послуг, а також помста та активізм.

1.2 Основні стадії реалізації DDoS атаки

Існує чотири основні етапи запуску DDoS-атаки:

1. Вибір агентів. Зловмисник вибирає агентів, які будуть виконувати атаку. Базуючись на природі виявлених вразливостей, більшість машин для використання в якості агентів були скомпрометовані.
2. Компрометація. Зловмисник використовує дірки у безпеці і вразливості апаратних машин для того, щоб впровадити код атаки. Зловмисник також вживає необхідних заходів для захисту коду від ідентифікації та деактивації. У прямій стратегії атаки DDoS, скомпрометовані вузли, так звані агенти, розташовані між зловмисником і жертвою. Самі того не підозрюючи, хости є спільниками, набраними з числа великої кількості незахищених хостів у Інтернеті з високошвидкісним підключенням. Стратегія атаки DDoS зазвичай є більш складною завдяки включенню проміжного шару вузлів між скомпрометованими комп’ютерами і жертвою. Це ще більше ускладнює відстеження шляху від жертви до зловмисників, головним чином, через складність розплутування інформаційного сліду за рахунок участі декількох машин або відстеження підключення через велику кількість розподілених маршрутизаторів або серверів. Якщо не використовується складний захисний механізм, для користувачів і власників скомпрометованих агентів важко усвідомити, що вони стали частиною системи DDoS атаки.
3. Зв'язок. Зловмисник спілкується з будь-якою кількістю обробників, щоб визначити, які агенти запущені, коли планувати атаки, або коли оновлювати агентів. Такий зв'язок між зловмисниками і обробниками може здійснюватися за допомогою різних протоколів, таких як ICMP, TCP або UDP. Виходячи з конфігурації мережі атаки, агенти можуть спілкуватися з одним обробником або декількома обробниками.

4. Атака. Зловмисник ініціює атаку. Жертва атаки, тривалість атаки, а також особливості атаки, такі як тип, довжина TTL і номери портів можна регулювати. Зловмисники використовують доступну пропускну здатність, і кожен з них посилає велику кількість пакетів на цільового хоста або мережу, щоб негайно перенавантажити їх ресурси.

1.3 Типи DDoS-атак

DDoS-атаки класифікуються різними дослідниками різними способами за різними критеріями. Наступні підрозділи містять типи DDoS-атак, засновані на рівнях семирівневої моделі OSI, підходи, використовувані для запуску атак, обсяг генерованого трафіку і на основі динаміки швидкості атаки.

1.3.1 DDoS атаки в залежності від рівня OSI

Атаки DDoS для конкретних рівнів, засновані на рівнях взаємоз'єднання відкритих систем (OSI), можна розділити на дві категорії: DDoS прикладного рівня і DDoS транспортного і мережевого рівнів. У атаці прикладного рівня зловмисник використовує рівень 7, тобто протоколи прикладного рівня, такі як HTTP і HTTPS, для передачі трафіку жертві. Такий трафік зазвичай відправляє на сервер серйозні запити для процесору і робить його зайнятим назавжди. Обсяг трафіку, необхідний для того, щоб сервер перестав відповідати на запити, є порівняно меншим, ніж обсяг іншого типу атаки. Трафік в атаці прикладного рівня не відрізняється від законного трафіку, що ускладнює його виявлення. У атаці на мережевому або транспортному рівнях зловмисник намагається вичерпати такі ресурси, як пропускна здатність або пам'ять таких пристроїв, як маршрутизатори, комутатори та брандмауери. Для досягнення цієї мети інфіковані машини передають жертві величезну кількість трафіку в рівнях 3 і 4. Така атака зазвичай велика в обсязі від декількох сотень Мбіт/с до декількох

сотень Гбіт/с. У таких атаках використовуються різні протоколи мережевого рівня, такі як протокол ICMP, протокол UDP і протокол керування передачею (TCP). Найбільш часто використовувані DDoS-атаки мережевого рівня - це затоплення TCP/SYN, ICMP echo, UDP flood, підсилення DNS і NTP.

1.3.2 Прямі і рефлекторні DDoS атаки

У DDoS-атаці не завжди скомпрометовані машини відправляють трафік жертві атаки. Сервери, що працюють на основі UDP-служб, часто використовуються зловмисниками для здійснення масових DDoS-атак. Такі сервери використовуються зловмисником як рефлектори. Виходячи з характеру атакуючих машин, DDoS-атаки поділяються на дві категорії, а саме: прямі і рефлекторні. У прямій атаці, яку зображено на рисунку 1.1, зловмисник використовує скомпрометовані машини безпосередньо для запуску DDoS-атак різних типів.

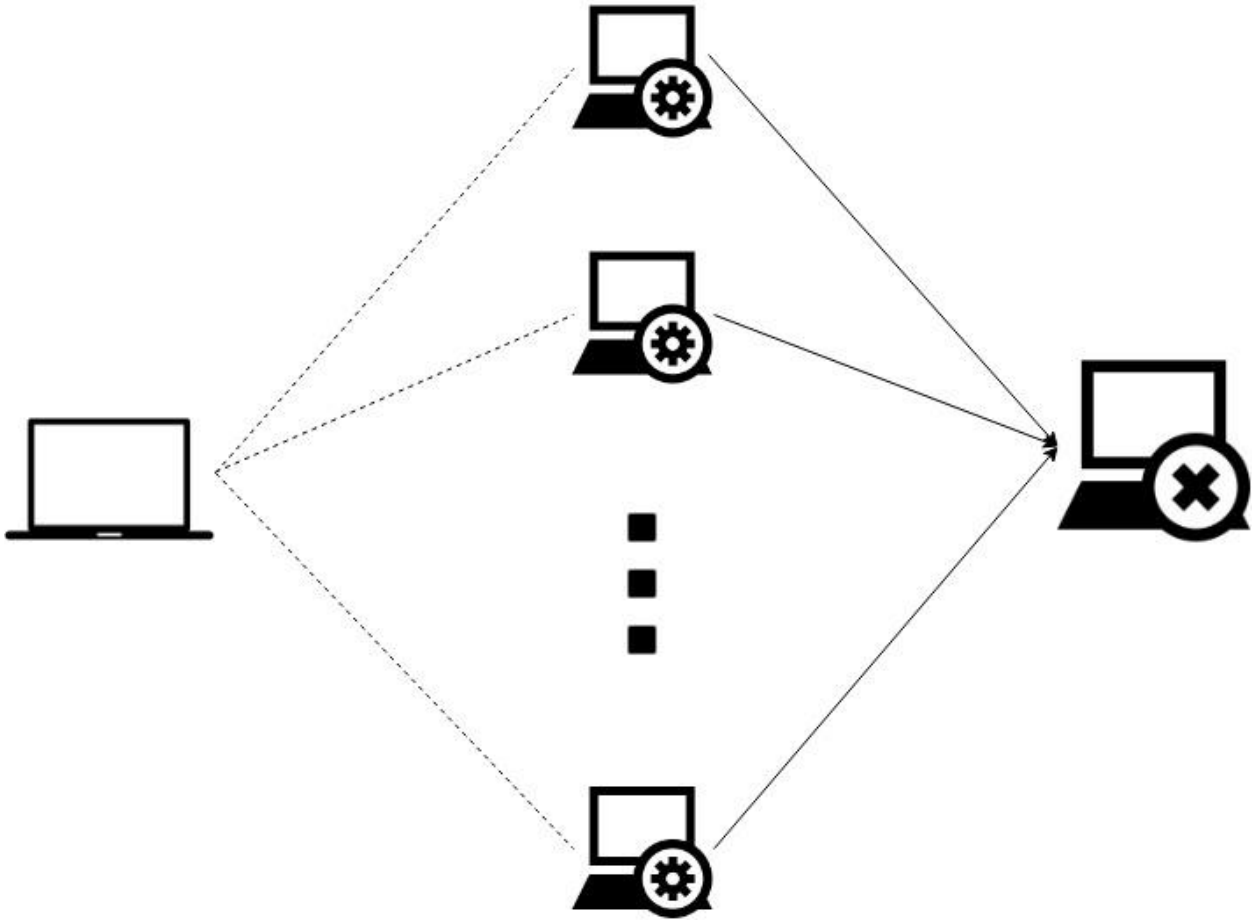

Рисунок 1.1 – Пряма DDoS атака

У той час як від рефлекторної атаки або посилення зображеної на рисунку 1.2, багато невинних проміжних вузлів, використовуються для генерування атаки. Зловмисник посилає запити на рефлекторні сервери, підмінивши IP-адресу джерела, як якщо б він був IP-адресом жертви. Як результат, ці сервери відповідають потерпілому, посилаючи повідомлення, обсяг яких зазвичай у багато разів перевищує початковий розмір повідомлення запиту. Отже, цей тип DDoS-атаки також називається атакою з підсиленням. Зловмисник використовує цю техніку, щоб посилити трафік атаки до декількох сотень разів. Підсилення атаки за допомогою DNS і NTP є прикладами DDoS-атак на основі відображення.[1]

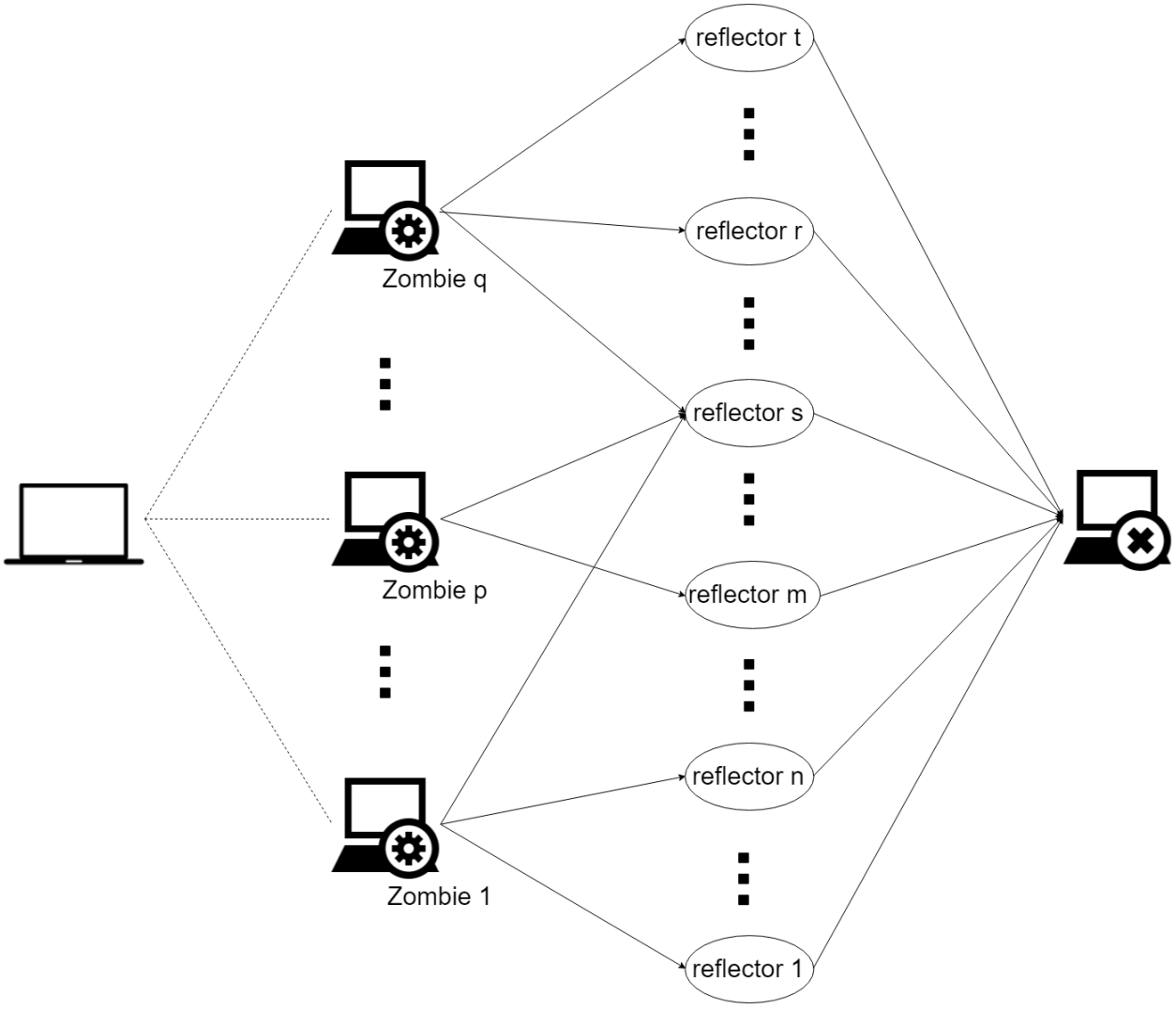

Рисунок 1.2 – Рефлекторна DDoS атака

1.3.3 Прямі та непрямі DDoS атаки

Можна також класифікувати DDoS-атаки на основі того, чи трафік атаки надсилається жертві безпосередньо або через посередників. У прямій атаці зловмисник посилає трафік атаки безпосередньо до жертви, використовуючи велику кількість скомпрометованих машин. У непрямій атаці зловмисник, замість того, щоб атакувати жертву безпосередньо, атакує інші сервіси, які важливі для того, щоб жертва залишалася функціональною. Атаки Link flooding, такі як crossfire[2] і coremelt[3], є прикладами непрямих DDoS-атак.

1.3.4 Потужні і малопотужні DDoS-атаки

DDoS-атаки також можуть бути класифіковані на основі обсягу трафіку атак, як малого, так і великого. У DDoS-атаці з низькою швидкістю, зловмисник зазвичай виконує атаку, відправивши трафік атаки за низькою швидкістю, що відповідає законному профілю трафіку. Наприклад, у випадку атаки прикладного рівня, зловмисник намагається вичерпати ресурси процесора жертви, надіславши запит, що інтенсивно використовує ресурси процесору. Аналогічно, в shrew attack, обсяг трафіку порівняно низький. У потужній DDoS атаці зловмисник посилає величезний обсяг трафіку на жертву. Це найпоширеніший тип DDoS-атаки. Великий об’єм трафіку, який іноді називають flash crowd, часто помилково приймають за DDoS-атаку, що призводить до скидання законних запитів користувачів. Однак flash crowd можна відрізнити від шкідливого трафіку, спостерігаючи за швидкістю введення нових адрес. У flash crowd раптово вводяться нові IP-адреси, що нагадують flooding attack, але швидкість введення нових IP-адрес скорочується через деякий час, хоча високий рівень запиту від нескомпрометованих користувачів може зберігатися. Прикладом flash crowd є проблеми Укрзалізниці у 2018 у зв’язку з Новорічними святами та великим притоком користувачів.

1.3.5 Типи атак на основі динаміки швидкості

На додаток до згаданої вище класифікації, DDoS-атаки можуть бути класифіковані на основі інших характеристик трафіку, таких як динаміка швидкості трафіку атаки. Класифікують DDoS-атаки на основі динаміки швидкості атаки на чотири категорії[1]:

1. Атака постійної швидкості: Швидкість атаки досягає максимуму протягом дуже короткого періоду часу. Всі скомпрометовані машини, отримавши команду від зловмисника, починають відправляти трафік з постійною

швидкістю. Цей тип атаки створює раптовий потік пакетів на стороні жертви.

2. Атака з підвищенням швидкості: Замість того, щоб атакувати жертву з повною силою миттєво, зловмисник поступово збільшує інтенсивність атаки. Зловмисник застосовує підвищення інтенсивності атаки, щоб зрозуміти реакцію жертви на атаку, для того, щоб уникнути механізмів виявлення жертви.
3. Пульсуюча атака: У такому вигляді атакуючий періодично активує групу ботів, щоб передати трафік жертві. Такий механізм використовується для того, щоб залишатися непоміченим для механізму виявлення. Shrew 52 є прикладом пульсуючої DDoS атаки, передаючи короткі синхронізовані пакети трафіку для зриву TCP-з'єднань на одній лінії, використовуючи слабкість механізму тайм-ауту повторної передачі TCP.
4. Атака підгрупами: Як і у випадку пульсуючої атаки, тут атакуючий також посилає імпульси трафіку жертві. Однак інфіковані машини поділяються на групи, і ці групи активуються і деактивуються в різних комбінаціях. Такий комбінований підхід атаки використовується зловмисником, щоб залишитися замаскованим і продовжувати атаку протягом більш тривалого періоду часу.

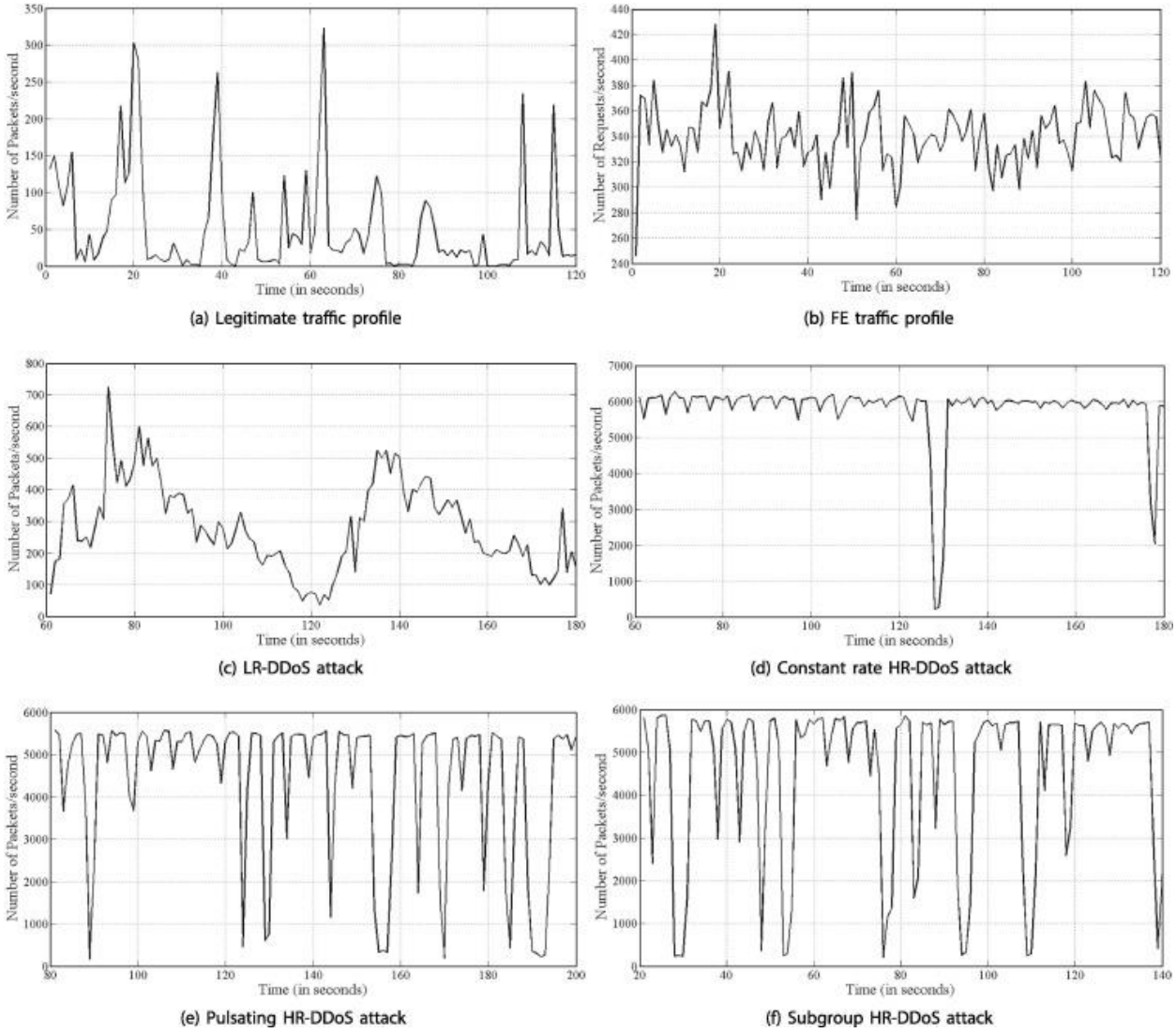

Рисунок 1.3 – Типи атак на основі динаміки [4]

1.4 UPnP

1.4.1 Огляд UPnP

Багато пристроїв і програм, які існують сьогодні, підтримують протокол UPnP (Universal Plug and Play). Протокол UPnP з'явився завдяки Microsoft на початку 1999 року, щоб принести концепцію plug and play. Ідея UPnP полягає в тому, щоб дозволити користувачеві підключити пристрій до локальної мережі, і цей пристрій буде просто працювати, незалежно від того, чи є пристрій принтером, сканером, файловим сервером або брандмауером. Вся конфігурація

прихована для користувача і замість цього виконується автоматично за допомогою самих пристроїв і програм.

Перша реалізація UPnP була у 2000 році. Windows XP також мав вбудовану підтримку UPnP з моменту її випуску у 2001 році. На даний момент існують реалізації для різних операційних систем, включаючи Windows, Linux і FreeBSD. Стек протоколів UPnP використовує чітко визначені Інтернет-стандарти, такі як HTTP, XML і SOAP.

1.4.2 Недоліки UPnP

У той час як UPnP зручна з точки зору користувача - програми, які повинні мати для переадресації динамічно виділені порти або потребують спеціальні порти на файрволі, можуть зробити це автоматично. Як визначено в поточному стандарті, UPnP не має механізму захисту за замовчуванням[5]. Існує функція доповнення, яка додає елементи безпеки, але не всі пристрої реалізують цю функціональність.

З поточним протоколом UPnP існує неявний довірчий зв'язок між усіма пристроями, що підтримують UPnP в одній мережі. Кожен пристрій є одноранговим і не існує механізму політики, щоб перевірити, чи дозволяється пристрою використовувати певну послугу.

Кілька простих скриптів, які реалізують лише невелику частину команд, що використовуються в UPnP, є достатніми в певних ситуаціях, щоб дозволити виставляти машини та служби у внутрішній мережі зовнішньому світу. Це можливо завдяки використанню пристрою, що використовує профіль UPnP Інтернет-шлюзу, і переадресацію портів на зовнішньому інтерфейсі пристрою шлюзу Інтернету до машин з внутрішньої мережі. Це робить внутрішню машину доступною для всіх у зовнішній мережі. Частина маршрутизаторів ADSL і бездротових точок доступу на ринку в даний час реалізують цей профіль, і багато з цих пристроїв є вразливими.

У специфікаціях пристрою UPnP конкретно для Інтернет-шлюзу згадується, що повинна бути можливість перенаправити порти на внутрішні багатоадресні та широкомовні адреси, тому пристрої можуть спільно використовувати потоки передачі/багатоадресної передачі, наприклад телевізійний потік. Перевагою для відправника та одержувача є те, що тільки один потік повинен бути відправлений до шлюзу, який буде повторно надісланий до його локальної мережі, так що він може бути розділений між машинами в локальній мережі, заощаджуючи пропускну здатність. Але переадресація портів на широкомовні адреси відкриває цілий ряд нових атак. Наприклад, Internet Printing Protocol[6], який використовується багатьма принтерами, використовує широкомовні адреси.

Відкриття та пересилання портів у брандмауерах через UPnP є серйозною загрозою. Спочатку може здаватися, що це може вплинути тільки на домашніх користувачів, але не на бізнес - ніякі продукти рівня підприємства, як правило, не підтримують UPnP, але, на мою думку, це твердження хибне. Це також може вплинути на бізнес-користувачів з ряду причин[5]:

- Багато малих підприємств підключені через звичайні лінії споживчого класу і мають такий же маршрутизатор, що і звичайні домашні користувачі.
- Бездротові точки доступу та маршрутизатори, які в першу чергу призначені для домашнього використання, також часто використовуються у малих компаніях. Багато адміністраторів не налаштовують належним чином точки бездротового доступу та маршрутизатори, так що, ймовірно, що UPnP також включений у цих мережах.
- Багато джерел, якщо не більшість, атак на мережі компаній є звичайними користувачами. Той факт, що мільйони маршрутизаторів з підтримкою UPnP були продані, говорить про те, що не слід ігнорувати дану вразливість.

Загроза для бізнесу полягає в тому, що сервіси, які не повинні піддаватися зовнішньому впливу, такі як внутрішні файли DNS або файлові сервери NFS / SMB, FTP тепер можна легко відкрити для всього світу. Ці файлові сервери часто містять важливу інформацію.

Можливо, багато людей не бачать загрозу в UPnP адресації портів, або що цей спосіб атаки занадто очевидний, і що цього недостатньо для її використання. Однак багато з найбільш ефективних атак реалізуються за допомогою простих вразливостей.

У січні 2013 року охоронна компанія Rapid7 в Бостоні повідомила про шестимісячну дослідницьку програму. Команда перевіряла сигнали від пристроїв з підтримкою UPnP, повідомляючи про їх доступність для підключення з Інтернету. 6900 мережевих продуктів із 1500 різних компаній і близько 81 мільйона IP-адрес відповідали на їхні запити. 80% пристроїв були домашніми маршрутизаторами; інші являлись принтерами, веб-камерами та камерами спостереження. Використовуючи UPnP-протокол, багато з цих пристроїв можуть бути доступними для зовнішніх користувачів [7].

1.4.3 Етапи роботи UPnP

В мережі пристроїв UPnP контрольні точки можуть виявляти пристрої, викликати дії служб пристрою і підписуватись на події. З іншого боку, пристрої реагують на викликані дії та надсилають події, коли змінюються змінні стану. Щоб зробити цю основну функціональність можливою, всі пристрої UPnP виконують однакові основні схеми або етапи роботи[5]:

- Адресація - пристрій приєднується до мережі, отримуючи унікальну адресу, яку інші користувачі можуть використовувати для зв'язку з нею.
- Опис - пристрій підсумовує свої послуги та можливості у стандартному форматі.
- Виявлення - пристрій знаходиться за допомогою контрольних точок, і інформація з опису пристрою надсилається на мережну одиницю, яка була ініціатором виявлення.
- Контроль - пристрій обробляє запити від контрольних точок для виклику дій.

- Події - служби пристрою повідомляють зареєстровані контрольні точки, коли відбуваються внутрішні зміни стану.
- Презентація - пристрій додатково надає адміністративний інтерфейс на основі HTML, що дозволяє здійснювати безпосередню маніпуляцію і моніторинг.

1.4.4 Адресація

Протокол IP є основним мережевим протоколом і є розумним вибором для підключення пристроїв UPnP. Сам по собі IP вимагає конфігурації, включаючи налаштування IP-адреси кожної кінцевої точки. Проте, відповідно до завдання конфігурації з нуля, ви не хочете, щоб користувачеві було потрібно вручну налаштувати кожен пристрій UPnP з його IP-адресою, але замість цього хочете механізм автоматичної адресації. Існує два протоколи адресації для вирішення цього конфлікту між необхідністю налаштування стека IP і бажанням уникнути конфігурації параметрів пристрою кінцевим користувачем: DHCP і Auto-IP. Кожен з них має свої сильні та слабкі сторони, і разом вони забезпечують додаткові рішення для адресації пристроїв UPnP[5].

Протокол динамічної конфігурації хоста (DHCP) надає структуру для передачі інформації про конфігурацію хостам по мережі TCP / IP, включаючи IP-адресу хоста, маску підмережі, шлюз за замовчуванням і сервер доменних імен. Це протокол клієнт / сервер, який використовує UDP як транспортний протокол. DHCP-клієнти надсилають повідомлення DHCP-серверам на порт 67 і отримують відповіді від серверів на 68 порту. DHCP-сервер керує пулом IP-адрес, автоматично призначаючи адреси мережним хостам, і повторно використовує адреси, які були звільнені. Використання сервера DHCP для призначення адрес кінцевим точкам IP централізує управління IP-адресами, гарантуючи, що кожна кінцева точка отримує унікальну адресу і уникає проблем, які виникають при

ручній конфігурації. Існують три механізми, які DHCP може використовувати для призначення IP-адрес клієнтам[8]:

- Автоматичне виділення. DHCP-сервер надає клієнту постійну IP-адресу.
- Ручне назначення. Мережевий адміністратор визначає призначення адреси для кожного хоста, і DHCP-сервер просто передає адресу клієнтам, коли вони запитують адресу від DHCP-сервера.
- Динамічне виділення. У цьому найпоширенішому режимі роботи сервер DHCP призначає IP-адресу клієнту протягом обмеженого періоду часу. Говорять, що клієнт має договір оренди за адресою. Після закінчення терміну оренди або виходу із мережі клієнта сервер може призначити цю адресу іншому клієнту.

Кожен пристрій UPnP повинен мати вбудований клієнт DHCP. Коли пристрій UPnP спочатку підключено до мережі, він шукає сервер DHCP для отримання IP-адреси. Теоретично, DHCP-сервер, що обслуговує мережу пристроїв UPnP, може використовувати будь-який з трьох механізмів. Тим не менш, динамічне виділення забезпечує найкращу відповідність, оскільки для кожного клієнта не потрібна конфігурація від адміністратора.

Використання DHCP для призначення адрес динамічно змінюваному набору пристроїв вимагає постійного доступу до серверу DHCP. Щоб працювати безперервно, сервер повинен знаходитися на машині, яка завжди включена, наприклад, шлюз Інтернету або домашній сервер ПК. Цілком можливо, що DHCP-сервер не може бути постійно ввімкненим, особливо в будинках або невеликих офісах без адміністративної підтримки. Відповідно до філософії конфігурації з нуля, розробники архітектури UPnP потребували іншого механізму адресації, щоб гарантувати, що пристрої UPnP могли отримувати адреси навіть у мережах без сервера DHCP. Авто-IP було рішенням.

Авто-IP - це метод, за допомогою якого кінцева точка в мережі IP може автоматично вибирати IP-адресу і маску підмережі за відсутності сервера DHCP. Авто-IP не замінює DHCP, але дає змогу клієнтам можливість автономного

налаштування, роблячи клієнтів більш надійними, дозволяючи їм отримувати адреси за відсутності служб DHCP. Пристрої UPnP використовують механізм Авто-IP, лише якщо DHCP-сервер відсутній, або якщо DHCP виходить з ладу. Крім того, архітектура пристрою UPnP визначає, що пристрій UPnP, який налаштував свою адресу за допомогою Auto-IP, повинен періодично перевіряти наявність сервера DHCP, щоб він міг переходити до адреси наданої DHCP.

Як тільки пристрій UPnP визначить, що він повинен використовувати Авто-IP для отримання IP-адреси, він починає з вибору адреси кандидата. Хоча фактичний алгоритм вибору адреси залежить від реалізації, адреса повинна входити в діапазон адрес, 169.254/16, які є приватними IP-адресами. Адреси в цьому діапазоні не перетинатимуть множину відкритих адрес. Після вибору адреси клієнт також налаштовує себе за допомогою маски підмережі класу B, за замовчуванням 255.255.0.0. Internet Assigned Numbers Authority (IANA) зарезервував цей діапазон для приватних IP-адрес, тому ніхто не може використовувати його в Інтернеті. Цей діапазон відомий як мережа LINKLOCAL. Також перший і останній із 256 адрес зарезервовані для майбутнього використання і не повинні бути вибрані.

Після того, як пристрій UPnP вибрав адресу, він повинен переконатися, що інший пристрій у мережі не використовує цю адресу. Для цього пристрій UPnP використовує Address Resolution Protocol (ARP) незвичним чином. ARP відображає будь-яку адресу мережевого рівня, на відповідну адресу каналу передачі данних, так що мережний стек може інкапсулювати IP-дейтаграму в кадрі Ethernet і відправляти його до адреси призначення. Мережевий хост після отримання запиту ARP перевіряє, чи має він IP-адресу, яка записана у запиті. Якщо це так, то хост відповідає на джерело запиту. Автор потім використовує Media Access Control (MAC) протокол для створення заголовка пакета, який він збирається надіслати хосту.

При спробі з'ясувати, чи використовується адреса в даний момент, пристрій UPnP, що самостійно налаштовується за допомогою функції Auto-IP, надсилає

запит ARP для вибраної адреси. Якщо жоден з хостів у мережі зараз не використовує IP-адресу, відповіді на повідомлення ARP не буде, і пристрій припускає, що він може вільно використовувати адресу.

Використовуючи Auto-IP для призначення адреси, хости можуть об'єднатися і сформувати тимчасову мережу без допомоги існуючої мережевої інфраструктури (DHCP і DNS-сервери). Ця функція забезпечує багато цікавих сценаріїв з пристроями UPnP. Наприклад, розглянемо двох користувачів з бездротовими пристроям, що зустрічаються в аеропорту і хочуть обмінюватися файлами. Якщо користувачі не знаходяться в безпосередній близькості від сервера DHCP, кожен пристрій призначає собі адресу з мережі 169.254/16. На цьому етапі кожен пристрій має мережеве підключення і може виявляти сервіси, такі як обмін файлами, які пропонує інший пристрій. Ця система пристроїв формує короткочасну мережу без підтримки будь-якої існуючої мережевої інфраструктури.

1.4.5 Виявлення

Як тільки пристрій UPnP отримав адресу, він готовий надати свої послуги контрольним точкам мережі. Виявлення, наступний етап роботи пристрою UPnP, дозволяє контролювати точки для пошуку пристроїв і послуг у мережі і знайти ті, які відповідають його критеріям пошуку. У цьому розділі спочатку розглядається виявлення служби в цілому, а потім розглядається Simple Service Discovery Protocol, протокол виявлення, що використовується пристроями UPnP.

Механізм виявлення мережевих сервісів повинен дозволити клієнтам запитувати мережу, щоб з'ясувати, чи наявні і доступні потрібні послуги в даний час, і повинні надавати клієнтам інформацію, необхідну для вибору конкретної послуги з набору доступних. Для цього пристрої та служби можуть інформувати про свою функціональність у мережі та надавати клієнтам інформацію про себе, щоб полегшити вибір клієнта. Пристрої та послуги інформують, надаючи клієнтам

інформацію про свою присутність та функціональність, яку вони надають. Клієнти, з іншого боку, виявляють і вибирають - знаходять потрібні їм послуги, а потім вибирають ті, які відповідають їхнім вимогам.

Simple Service Discovery Protocol (SSDP) був розроблений як просте рішення для виявлення ресурсів на основі протоколу HTTP у локальній мережі, що не потребує конфігурації, керування або адміністрування. SSDP не намагається вирішити проблему виявлення ресурсів на базі Інтернет-протоколу HTTP. Ця проблема залишається для інших протоколів, таких як Universal Description Discovery and Integration (UDDI). SSDP використовує децентралізований підхід до виявлення послуг, тобто ніхто, крім пристрою, не зберігає інформацію про ресурси, їх місцезнаходження та їх доступність. Замість цього кожен клієнт безпосередньо запитує мережу, і кожен ресурс відповідає безпосередньо на ці запити.

Децентралізований пошук є надійним, оскільки він не вимагає будь-яких централізованих точок або конфігурації адміністратором мережі. Ця інформація завжди актуальна, оскільки ресурси відповідають безпосередньо на запити та видають клієнтам оновлення щодо їх статусу. Однак ця система вимагає, щоб кожен ресурс прослуховував і обробляв запити на виявлення. Оскільки кількість пристроїв у мережі зростає, цей підхід стає менш привабливим - інформація про стан дублюється для всіх клієнтів, збільшується пропускна спроможність мережі, яка витрачається на трафік виявлення, а обчислювальна потужність витрачається даремно, оскільки кожен пристрій займається прослуховуванням та обробкою повідомлень виявлення.

Існує два типи запитів SSDP. Перші, запити на виявлення, дозволяють клієнтам SSDP шукати пристрої, які використовують SSDP. Друге, оголошення про наявність, дозволяє пристроям, які використовують SSDP, оголосити про свою присутність у мережі. Баланс запитів виявлення SSDP і оголошень про присутність призначений для того, щоб зробити протокол ефективним, зменшуючи мережевий трафік. Коли ресурс з'являється в режимі онлайн, він

оголошує про свою присутність. Це дозволяє всім клієнтам дізнатись, що ресурс доступний. З цього моменту ресурс не повинен надсилати будь-які інші повідомлення про присутність, за винятком випадків, коли пристрій виходив із мережі. Будь-які клієнти, які з'являться в режимі онлайн після того, як ресурс оголосив про свою присутність, розсилають запити на виявлення. Якщо ресурс підтримує запитану послугу, він реагує на клієнта. Клієнту не потрібно повторювати запит на виявлення, тому що будь-які ресурси, що з’являються в мережі після того, як він надсилав запит, оголосять свою присутність. Результатом є те, що ні клієнт, ні сервер не повинні надсилати постійні потоки повідомлень.

Оскільки SSDP стосується виявлення веб-ресурсів, доцільно розглядати HTTP як транспорт. Проте, HTTP працює через TCP, надійний, рівноправний протокол. Оскільки SSDP вимагає відправлення запитів на виявлення та повідомлень про присутність кожному вузлу в локальній мережі, використання прямих підключень HTTP до кожного користувача мережі неможливо. Замість цього, SDDP використовує HTTP над багатоадресною UDP для відправки повідомлень кожному вузлу SSDP в мережі. SSDP-клієнти здійснюють груповий запит виявлення HTTPMU на адресу 239.255.255.250:1900, яка є зарезервованою груповою адресою та портом SSDP, що має локальну адміністративну область, що обмежує доставку групової передачі в адміністративний домен. Не передбачено забезпечення надійності повідомлень про виявлення, переданих через UDP, протокол дейтаграм, який не забезпечує гарантії надійності. Служби SSDP прослуховують груповий канал SSDP, щоб почути запити на виявлення. Якщо ресурс SSDP приймає запит виявлення HTTPMU, який відповідає запропонованій службі, він відповідає, надіславши відповідь безпосередньо клієнту SSDP, який видав пошук, використовуючи HTTPU.

Запит на виявлення SSDP вводить метод запиту HTTP, MSEARCH і заголовок для ідентифікації мети пошуку ST. Клієнт встановлює ST на URI потрібного типу послуги. Заборонено використовувати декілька заголовків ST.

Більш складні пошуки, такі як визначення логічних операцій або пар імен/значень, не підтримуються. SSDP підтримує лише базові пошуки, щоб зберегти його простим у впровадженні і простим протоколом виявлення. У таблиці 1.1 наведено список заголовків, які використовуються у запиті UPnP SSDP.

Таблиця 1.1 — Список заголовків у запиті UPnP[5]

Заголовок	Обов’язковий	Тип	Опис
Host	так	Ім'я домену або IP-адреса та додатковий порт	Якщо порт не вказано, передбачається порт 80. Для запитів на виявлення UPnP це значення "239.255.255.255:1900".
Man	так	Має бути ssdp:discover	Клієнт SSDP встановлює заголовок Man "ssdp: discover", щоб вказати, що це повідомлення виявлення SSDP, яке має зрозуміти одержувач.
MX	так	Ціле число	Максимальна кількість секунд для відповіді. Щоб зменшити навантаження на клієнта, що приймає багато відповідей на запит, відповідач посилає відповідь у випадковій точці протягом цього інтервалу часу.
ST	так	URI	Ціль пошуку для повідомлення виявлення Це те, що шукає клієнт.

Після того, як пристрій, що його намагаються виявити, він відповідає повідомленням безпосередньо відправнику. Окрім надання типу послуги та

унікального імені служби, результати виявлення надають інформацію про закінчення терміну дії та місцезнаходження. У таблиці 1.2 перелічені заголовки, які використовуються в повідомленнях відповіді на виявлення UPnP SSDP.

Таблиця 1.2 — Список заголовків у відповіді UPnP[5]

Заголовок	Наявність	Тип	Опис
Cache-control	Обов’язкова	різний	Існують різні налаштування кеш-керування, описані в RFC 2068, «Протокол передачі гіпертексту - HTTP/1.1». Відповіді на виявлення UPnP використовують max-age = секунди, щоб визначити, через скільки часу оголошення буде закінчуватися.
Date	Рекомендовано	RFC1123 date	Коли відповідь була згенерована
Location	Обов’язкова	URL	Значенням є URL-адреса до документа опису кореневого пристрою.
Ext	Обов’язкова	Нема значення	Підтверджує, що заголовок Man у запиті (ssdp: find) був зрозумілий.
Server	Обов’язкова	Текстовий рядок	Конкатенація назви ОС, версії ОС, UPnP / 1.0, назви продукту та версії продукту, що визначається постачальником UPnP.
ST	Обов’язкова	URI	Мета пошуку для повідомлення виявлення. Відповідь містить те ж значення ST, що і відповідне повідомлення запиту.
USN	Обов’язкова	URI	Це унікальне ім'я служби для виявленого пристрою.

Для пристроїв UPnP поле USN може мати різні форми залежно від типу запиту у вихідному заголовку ST, як показано у таблиці:

Таблиця 1.3 — Список полів USN у відповіді UPnP[5]

Форма	Опис
uuid:device-UUID:upnprootdevice	Ця форма використовується, коли запит вказаний upnp-rootdevice. На цей тип пошуку може відповідати кожен пристрій UPnP.
uuid:device-UUID	Ця форма використовується, коли певний пристрій здійснює пошук за ідентифікатором пристрою. На подібне повідомлення має бути не більше однієї відповіді.
uuid:deviceUUID:urn:schemas-upnporg: device:deviceType:ver	Ця форма використовується у відповідь на пошук конкретного виду пристрою, наприклад, Інтернет-шлюзу. Всі шлюзові пристрої відповідатимуть набором заголовків USN у цьому форматі.
uuid:device-UUID:urn:schemas-upnporg:device:serviceType:ver	Ця форма використовується у відповідь на пошук певного виду послуг, наприклад, сервіс пристрою відтворення медіа. Багато різних типів пристроїв можуть включати один і той же тип послуг. Всі екземпляри відповідатимуть у цій формі.

У відповідь на запит на виявлення можуть виникнути наступні відповіді: відповіді на виявлення, подібні до наведених у таблиці, починаються з типової HTTP-відповіді, за яким слідують кілька додаткових заголовків для передачі інформації

про виявлений ресурс. Заголовок Location містить адресу, за якою клієнт може реально використовувати виявлений пристрій.

Заголовок Ext повертається без значення, щоб вказати, що одержувач зрозумів обов'язковий заголовок у запиті. Оскільки оригінальний запит на пошук шукав усі пристрої шлюзу Інтернету, USN використовує форму, яка включає унікальний ідентифікатор відповідного пристрою, об'єднаний з вихідною ціллю пошуку, urn: schemas-upnp-org: device: InternetGatewayDevice: 1. Заголовок USN містить ідентифікатор відповідного ресурсу. Заголовок Server надає інформацію про операційну систему пристрою, назву продукту та версію. Заголовок Cache-Control повідомляє запитувачу, як кешувати повернуту інформацію.

Заголовок ST містить те ж значення ST, що і вихідний запит. Це те, що шукав запитувач. У цій відповіді немає жодного асоційованого вмісту, тому заголовок Content-Length встановлюється рівним 0. Існує декілька правил, яким повинні відповідати відповіді на виявлення SSDP:

- Тільки послуги SSDP з типом служби, які відповідають значенням у заголовку ST, можуть відповідати на запит ssdp:discover на груповому каналі SSDP.
- Успішна відповідь на запит ssdp:discover повинна включати заголовки ST та USN.
- Відповіді на ssdp:discover запити, надіслані по каналу групової передачі SSDP, повинні бути надіслані до тієї ж IP-адреси та порту, який зробив ssdp:discovery запит.
- Відповідь на запит ssdp: discovery повинна включати адресу служби, виражену через заголовок Location. Інформація про місцезнаходження визначає, як слід звертатися до певної служби. Один або більше URI адреси можуть бути включені у відповідь на виявлення.
- Відповідно до проекту стандарту SSDP, відповіді на ssdp:discovery запити повинні містити заголовок Cache-Control: max-age або Expires. Коли вони присутні, вони обробляються в порядку, визначеному HTTP / 1.1, тобто

заголовок Cache-Control має перевагу над заголовком Expires. Якщо у відповіді на запит ssdp:discovery не надається заголовок Cache-Control або Expires, інформація, що міститься в цій відповіді, не повинна кешуватися клієнтами SSDP.

- Інформація про термін дії ідентифікує, як довго клієнт SSDP повинен зберігати інформацію про послугу в своєму кеші. Після закінчення терміну дії запису його потрібно видалити з кешу клієнта SSDP.

1.4.6 Опис

Описання пристроїв і служб - це просто документи XML, які відповідають мові шаблонів UPnP, синтаксису XML, визначеному форумом UPnP для створення описів пристроїв і послуг. Ця основна мова шаблону використовується різними робочими комітетами Форуму UPnP для визначення стандартних пристроїв і послуг, які вони повинні містити. Робочі комітети UPnP Форуму починаються з мови шаблонів UPnP і створюють шаблони опису документів для конкретного типу пристрою та його послуг. При впровадженні пристроїв UPnP постачальники пристроїв заповнюють шаблони опису документів, надаючи інформацію про постачальника. Концептуально, шаблон пристрою UPnP визначає тип пристрою, тоді як документ опису пристрою створює шаблон з інформацією, наданою постачальником.

Шаблони пристроїв і служб містять інформацію про стандартний пристрій і його послуги, дії, параметри, змінні тощо. Наприклад, шаблони пристроїв і послуг для пристрою шлюзу Інтернету є стандартизованами. Постачальники, що впроваджують сумісні зі стандартами пристрої, починають з цих шаблонів, визначених робочими комітетами, і заповнюють специфічну інформацію про постачальників, можливо, диференціюючи свої пристрої за допомогою додаткових послуг, розширення існуючих послуг або вбудовування додаткових пристроїв. Отриманий документ опису, повернутий з конкретного пристрою

UPnP, таким чином, відповідає визначеному UPnP Форуму синтаксису, реалізує типи пристроїв і послуг, визначені робочим комітетом Форуму UPnP, і включає в себе інформацію, специфічну для постачальника і обладнання.

Існує декілька правил, що регулюють документи опису UPnP[5]:

- Усі елементи та атрибути чутливі до регістру. Всі інші значення, крім URL-адрес, не чутливі до регістру.
- Порядок елементів не є важливим. Елементи можуть бути в будь-якому порядку, не змінюючи значення опису документа.
- Необхідні елементи повинні бути записані один раз без дублікатів.
- Рекомендовані або додаткові елементи можуть виникати не більше одного разу.
- Як зазначено у Flexible XML Processing Profile (FXPP), контрольні точки повинні ігнорувати будь-які невідомі елементи та їх під-елементи або вміст, а також будь-які невідомі атрибути та їх значення при обробці описів пристроїв і послуг.
- Символ амперсанда (&) не дозволено в XML. Якщо потрібно, він повинен бути перетворений у &amp.
- Бінарні данні не можуть бути безпосередньо включені в документ XML. Спочатку його слід перетворити на текст, використовуючи такі формати, як base64 або binhex. До двійкових данних можна посилатися опосередковано, використовуючи URL-адресу данних у документі.
- Пристрої, стандартизовані робочими комітетами UPnP Форуму, повинні мати цілий номер версії. Пізніші версії повинні бути наборами, які включають в себе попередні версії.

1.4.7 Контроль

Simple Object Access Protocol (SOAP) - це протокол, який об'єднує XML і HTTP, щоб забезпечити механізм обміну повідомленнями та віддалених

процедур. XML використовується для передачі вмісту повідомлень, тоді як HTTP використовується для передачі повідомлень до місця призначення. Цей підрозділ описує SOAP в цілому[5].

SOAP визначається як набір умовностей, які регулюють формат і правила обробки повідомлень SOAP. SOAP складається з чотирьох частин[5]:

- конверт SOAP. Схема XML, яка визначає рамки для опису того, що знаходиться в повідомленні, як його обробляти, і чи є воно необов'язковим або обов'язковим.
- Правила кодування SOAP. Інша схема XML, яка визначає набір правил для вираження екземплярів типів данних, що визначаються програмою.
- Зв'язування SOAP. Угода про використання різних транспортних протоколів. SOAP потенційно може використовуватися в поєднанні з безліччю інших транспортних протоколів. (Однак SOAP найчастіше передається за допомогою HTTP.)
- Представлення RPC SOAP. Угода для представлення віддалених викликів процедур і відповідей.

Повідомлення SOAP є основною одиницею зв'язку між пристроями. SOAP-повідомлення написані в XML, тому SOAP платформа є незалежною (будь-яка система, здатна створювати і аналізувати XML-документи, може надсилати і отримувати повідомлення SOAP). Через потужність XML, повідомлення SOAP можуть бути досить складними за структурою і можуть передавати дуже складні типи данних.

1.4.8 Події

Загальна архітектура сповіщень про події (General Event Notification Architecture) - це система видавця/абонента, за допомогою якої абонент може запитувати, поновлювати або скасовувати підписку. Абонент спочатку надсилає повідомлення про підписку видавцеві. Якщо підписка прийнята видавцем, вона

відповідає ідентифікатором підписки та тривалістю для цієї конкретної підписки. Подальші операції з підписки, такі як поновлення та скасування, використовують ідентифікатор підписки для посилання на підписку. Щоб продовжити підписку, перед тим, як підписка закінчиться, абонент надсилає повідомлення про оновлення. Коли абонент більше не зацікавлений в отриманні подій від видавця, він може скасувати підписку. Підписки також можуть бути скасовані видавцем.

GENA запроваджує три методи HTTP, які використовуються для керування підписками на події та доставку повідомлень[5]:

- SUBSCRIBE, щоб підписатися на отримання сповіщень про події та відновити існуючу підписку. Заголовки будуть різними залежно від того, яка функція призначена.
- UNSUBSCRIBE для припинення підписки.
- NOTIFY надсилати повідомлення про подію абоненту.

GENA вводить наступні заголовки, які використовуються з новими методами HTTP:

- CALLBACK використовується для передачі URL-адреси, яка буде використана для виклику іншого об'єкта. Наприклад, під час реєстрації абонент відправляє заголовок CALLBACK для отримання подій. Видавець використовує цю URL-адресу для надсилання сповіщень про події.
- NT - тип повідомлення. Вона використовується для того, щоб повідомити абонента, яке це повідомлення.
- NTS є підтипом сповіщення. Це дозволяє додатково покращити тип сповіщення.
- SID є ідентифікатором підписки. Цей ідентифікатор створюється видавцем для посилання на підписку. І видавець, і абонент використовують цей ідентифікатор при спілкуванні з іншим і посилаються на певну підписку.

Повідомлення GENA також використовують стандартні заголовки HTTP, такі як Host, Timeout, Date, Server, Content-Length і Content-Type.

Концептуально проста модель видавця/абонента GENA легко відображає об'єктну модель UPnP: контрольні точки UPnP є абонентами, тоді як послуги UPnP є видавцями. Сам пристрій UPnP не є джерелом подій, а просто контейнер для послуг. Архітектура пристрою UPnP встановлює додаткові умови для подій поза базовими можливостями, наданими GENA[5]:

- опис служби UPnP включає в себе список дій, на які реагує служба, і список змінних, що модифікують стан служби під час виконання. Будь-яка з цих змінних може бути ідентифікована як здатна до пошуку подій при зміні стану. Якщо одна або більше з цих змінних стану змінюється, то контрольні точки можуть реєструватися для отримання подій від служби, і служба публікує повідомлення про події, коли будь-яка з цих змінних змінюється.
- Повідомлення про події на основі XML. Служба вказує зміни до змінних стану шляхом надсилання повідомлень про події до контрольних точок. Повідомлення про події містять назви однієї з змінних стану і поточне значення цих змінних, виражене за допомогою синтаксису XML, мови шаблонів UPnP для запису подій.
- Повідомлення про початкову подію. Повідомлення про спеціальні події надсилаються, коли контрольна точка спочатку підписується на отримання подій з сервісу послуг. Це спеціальне перше повідомлення включає в себе імена та значення для всіх подій, що надаються службою. Це повідомлення дозволяє абоненту ініціалізувати свою модель стану сервісу.
- Всі абоненти отримують всі повідомлення про події. Збігання в архітектурі UPnP розраховане на те, щоб всі абоненти були однаково поінформовані про наслідки будь-яких дій. Всім абонентам надсилаються повідомлення про події, і кожне повідомлення про подію містить значення всіх подій. Не передбачено механізму підписки на повідомлення про події на основі змінної.
- Як механізм виявлення помилок, який гарантує, що абоненти отримали всі відправлені повідомлення про події, видавець реалізує окремий ключ події

для кожної підписки. Для кожного повідомлення про подію абоненту, видавець збільшує ключ події і включає значення в сповіщенні. Для утримання цього значення архітектура UPnP вводить заголовок SEQ, що використовується в повідомленні про подію.

1.4.9 Презентація

У пристроях UPnP є вбудовані веб-сервери, оскільки більшість протоколів зв'язку, які вони використовують, включаючи опис пристроїв на основі XML, повідомлення керування SOAP та повідомлення, пов'язані з подіями GENA, використовують HTTP. Крім використання внутрішнього веб-сервера для програмного керування, пристрої також можуть використовувати вбудований веб-сервер із веб-інтерфейсом для управління пристроєм. Наприклад, замість контрольної точки, яка видає запит SOAP і отримує відповідь, адміністратор, який використовує веб-браузер, може завантажити певну URL-адресу і переглянути інформацію про пристрій або керувати нею за допомогою форми, сформованої на веб-сервері.

Веб-інтерфейс дозволяє адміністратору забезпечити правильне функціонування пристрою, а також діагностувати та усунути проблеми з пристроєм. Зокрема, адміністратор може використовувати веб-інтерфейс пристрою для:

- Маніпулювання робочими параметрами пристрою.
- Перегляд статистики пристрою.
- Ручний виклик дій служб пристрою.

1.5 Підміна IP-адреси

Підміна IP-адреси - це створення IP-пакетів з використанням чужих IP-адрес. Ця методика використовується з очевидних причин. Розглядаючи IP-

заголовок, можна побачити, що перші 12 байт містять різну інформацію про пакет. Наступні 8 байт містять IP-адреси джерела і призначення. Використовуючи один з декількох інструментів, зловмисник може легко змінити ці адреси - зокрема, поле "вихідна адреса". Поширеною помилкою є те, що підміни IP-адрес можна використовувати для приховування нашої IP-адреси під час серфінгу в Інтернеті, спілкування в Інтернеті, надсилання електронної пошти тощо. Як правило, це не так. Підробка IP-адреси джерела призводить до того, що відповіді будуть неправильно спрямовані, тобто ви не зможете створити нормальне мережеве з'єднання.

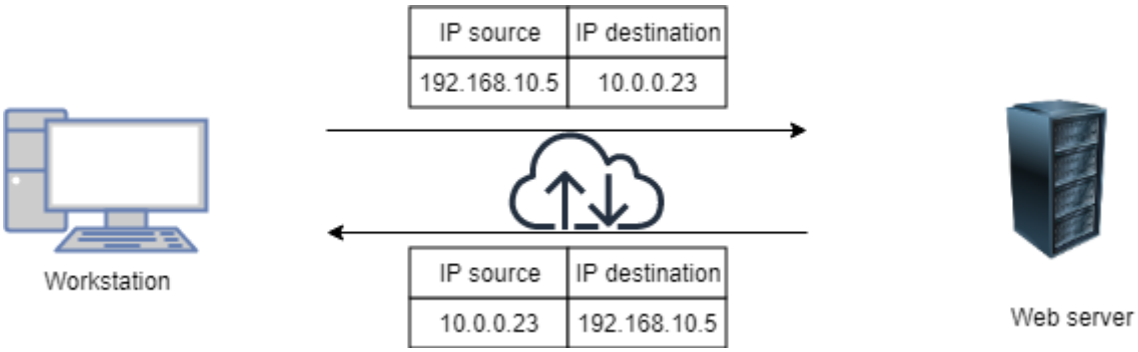

Рисунок 1.4 – Взаємодія між сервером та робочою станцією

Рисунок 1.4 ілюструє типову взаємодію між робочою станцією з IP-адресою джерела, що надсилає запити веб-серверу. Коли робоча станція запитує сторінку з веб-сервера, запит містить IP-адресу робочої станції (тобто IP-адресу джерела 192.168.10.5) і адресу веб-сервера, який надсилає відповідь (тобто IP-адреса призначення 10.0.0.23). Веб-сервер повертає веб-сторінку, використовуючи IP-адресу джерела, вказану в запиті, як IP-адресу призначення (192.168.0.59) та власну IP-адресу як вихідну IP-адресу (10.0.0.23).

Підміна IP-адреси джерела на рисунку 1.5 ілюструє взаємодію між робочою станцією, що запитує веб-сторінки, використовуючи підроблену IP-адресу джерела і веб-сервер, який надсилає відповіді. Якщо робоча станція використовує підроблену IP-адресу джерела (тобто 172.16.0.6), веб-сервер, який виконує запит

на веб-сторінку, спробує надіслати відповідь на ту IP-адресу, що вважається вихідною (тобто на робочу станцію 172.16.0.6). Робоча станція з підробленою IP-адресою отримуватиме несанкціоновані спроби встановлення зв'язку з веб-сервером, які вона просто відкине.

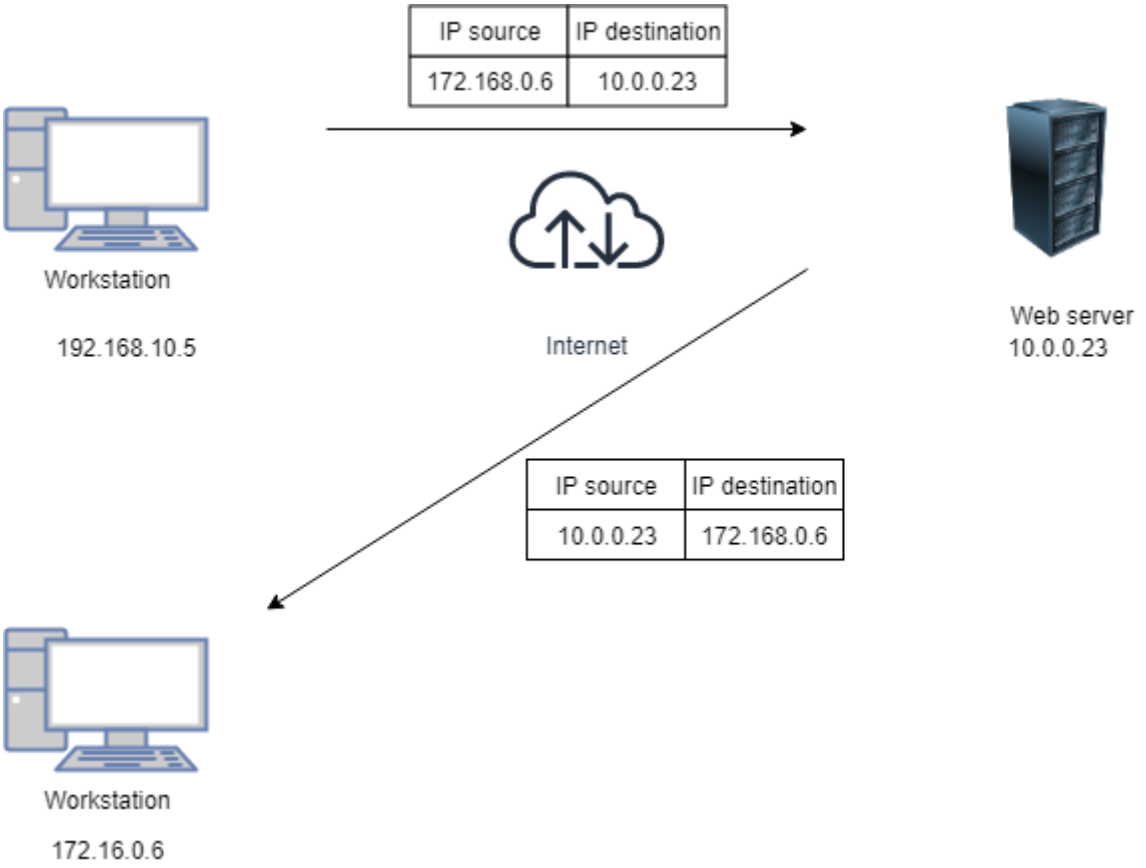

Рисунок 1.5 – Підміна IP-адреси при взаємодії з сервером

Механізм IP-маршрутизації – надсилання пакету від точки до точки до того моменту поки пакет не надійде до кінцевої точки. Кожен пакет IP маршрутизується окремо. Маршрут IP-пакета визначається всіма маршрутизаторами, через які проходить пакет. Підробка IP-адреси можлива тому, що маршрутизатори вимагають лише перевірки IP-адреси призначення в пакеті для прийняття рішення про маршрутизацію. IP-адреса джерела не вимагається маршрутизаторами, а недопустима IP-адреса джерела не впливатиме на доставку

пакетів. Ця адреса використовується лише машиною призначення, коли вона відповідає на адресу відправника.

Цей тип атаки найбільш ефективний там, де між машинами існують довірчі відносини. Наприклад, у деяких корпоративних мережах часто існує довіра між внутрішніми системами, завдяки чому користувачі можуть входити в систему без імені користувача або пароля, якщо вони підключаються з іншої машини у внутрішній мережі. Шляхом підробки з'єднання з довіреною машиною зловмисник з тієї ж мережі може отримати доступ до цільової машини без автентифікації. Підміна IP-адрес найчастіше використовується в атаках відмови в обслуговуванні, де мета – перенавантаження ресурсів цілі обсягом трафіку, а зловмисник не зацікавлений в отриманні відповідей на пакети. Пакети з підробленими IP-адресами важче фільтрувати, оскільки кожен підроблений пакет виходить з іншої адреси, і вони приховують справжнє джерело атаки.

Атаки відмови в обслуговуванні, які використовують підміну адрес, зазвичай випадково вибирають адреси з усього IP-адресного простору, хоча більш складні механізми підробки можуть уникнути неперехідних адрес або невикористаних частин простору IP-адрес. Розповсюдження великих бот-мереж робить підміни менш важливими при атаках відмови в обслуговуванні, але зловмисники, як правило, мають підміну адрес як один із інструментів. Вони можуть використовувати цей інструмент, тому захист від атак відмови в обслуговуванні, які покладаються на вірність вихідної IP-адреси в пакетах атаки може мати проблеми з підробленими пакетами. Під час даної атаки ми використали підміну IP-адреси.

Висновки до розділу 1

В даному розділі були розглянуті теоретичні відомості щодо реалізації DDoS атак з використанням різних технік, а також детально розібрано взаємодію користувача з UPnP набором протоколів.

Даний розділ сформував глибокі знання для подальшого експериментального дослідження вразливості SSDP протоколу та методу виправлення цієї вразливості. В нашому випадку потрібно бути особливо уважним до вибору методу захисту, оскільки вирішення проблеми може зробити неможливим використання повного функціоналу набору протоколів, що в свою чергу негативно вплине на враження користувача від використання кінцевого продукту.

2 СТАН ЗАХИЩЕНОСТІ ТА ВІДОМІ АТАКИ ПОВ’ЯЗАНІ З SSDP

Вектори атаки посилення є одними з найбільш часто використовуваних інструментів в арсеналі зловмисника DDoS. В останньому кварталі 2017 року ми побачили, що підсилення NTP використовувалося приблизно на 33% всіх DDoS нападів на наших клієнтів, в той час як DNS і SSDP підсилювали вектори відігравали роль у 17% і 13,7% атак, відповідно. Для зловмисників, вектори посилення пропонують можливість запуску навантажень, занадто великих для пропускної здатності, без необхідності настільки ж великих ресурсів ботнету. З точки зору пом'якшення, вони являють собою меншу загрозу, оскільки, на сьогоднішній день, більшість послуг з ліквідації наслідків позбулись пункту, де пропускна здатність атаки є головним завданням, або будь-якою іншою проблемою. Більш важливим є те, що заголовки вихідних портів корисних навантажень підсилення слідують передбачуваному шаблону, що полегшує їх фільтрування на межі мережі. Наприклад, блокування всіх пакетів з вихідним портом 53 вважається перевіреним методом для пом'якшення атак підсилення DNS. Нещодавно, пом'якшуючи атаку з посиленням SSDP, ми бачили докази корисного навантаження з нерегулярними даними порту джерела, що мало хто в нашій галузі вважає можливим[9].

Пристрої UPnP можуть використовуватися для заплутування данних порту джерела посилення. Зокрема, метод ухилення не обмежується посиленням DNS, оскільки наш наступний тест показав, що він ефективний для SSDP, DNS і NTP-атак. Крім того, немає підстав вважати, що інші вектори ампліфікації не будуть працювати так само добре. Це призводить до великої зміни парадигми в тому, як сьогодні пом'якшуються атаки з посиленням. Оскільки вихідна IP-адреса та інформація про порти більше не слугують надійними факторами фільтрації, найбільш імовірною є відповідь на те, щоб виконати глибоку перевірку пакетів для визначення корисних навантажень підсилення - більш ресурсомісткий процес, який є складним завданням.

PLXsert виявив, що 4,1 мільйона UPnP-пристроїв, що стикаються з Інтернетом, є потенційно вразливими до використання цього типу DDoS-атаки. Це становить приблизно 38 відсотків з 11 мільйонів знайдених пристроїв UPnP. Розподіл цих пристроїв по всьому світу, показаний на малюнку 10 і малюнку 11. Цей обсяг і розподіл створює виклик для пом'якшення, керування виправленнями, оновлення і очищення. Поширеність уразливих пристроїв, швидше за все, призведе до розробки нових інструментів, щоб скористатися перевагами протоколів SSDP і SOAP, які, ймовірно, також призведуть до інструментів посилення атаки на основі пристроїв UPnP і ботнетів, що користуються широким попитом на тіньовому ринку DDoS[10].

Рисунок 2.1 – Розповсюдження вразливих UPnP пристроїв

Під час повторення цього типу атаки в локальному лабораторному середовищі PLXsert апроксимував коефіцієнт посилення атаки і визначив посилення приблизно в 33%. Першою компанією, яка оприлюднила данні з даного типу атаки, була компанія Akamai. Під час атаки на клієнта Akamai була залучена

велика кількість UPnP пристроїв. Пік трафіку 54,35 гігабіт в секунду і 17,85 мільйона пакетів в секунду. У момент проведення дослідження PLXsert виявив, що 4,1 мільйона пристроїв для доступу до Інтернету в Інтернеті потенційно вразливі до того, що можуть бути використані в цьому типі DDoS-атаки. Це становить приблизно 38 відсотків з 11 мільйонів знайдених пристроїв UPnP. Поширеність уразливих пристроїв, швидше за все, призведе до розробки нових інструментів, які, ймовірно, також призведуть до виникнення інструментів посилення атаки на основі пристроїв UPnP[10].

У 3-му кварталі ми спостерігали напади, коли зловмисники вводили невеликі обсяги неправильно сформованих пакетів в потік данних для маскування. Отже, трафік атаки в просторі кожної IP-адреси був достатньо малий, щоб обійти виявлення, але достатньо великим, щоб нанести шкоду цільовому сайту або навіть всій мережу інтернет провайдера. Внаслідок незначного розміру шкідливого трафіку, типові пристрої безпеки, розгорнуті інтернет провайдерами на 3-4 рівнях OSI, не можуть виявити шкідливий трафік, перш ніж він завдасть шкоди. Це відбувається тому, що поріг виявлення значною мірою базується на обсязі заголовка пакету для цільових IP-адрес. Bit-and-piece використовує великі поверхні атаки на інтернет провайдерів на рівнях 3-4 моделі OSI, тоді як традиційні атаки спрямовані на один або кілька IP-адрес, які є критично важливими службами, такі як веб-сайти і поштові сервери, і перевантажують цілі, посилаючи великі об’єми данних. Оскільки показник трафіку є значним і атака очевидна, порівняно легко виявити аномалії та пом'якшити традиційні об'ємні атаки. У більшості випадків провайдери з можливостями пом’якшення атак будуть поглинати більшу частину впливу від цих атак, хоча і не 100%. Вектори атаки, виявлені через мережу Nexusguard, показують, що атаки посилення були домінуючим у третьому кварталі 2018. Використання посилення атаки через SSDP складає складає 94,1%. Інші методи посилення зображені на рисунку 2.2. Вихідні дані показують, що інтернет провайдери на рівнях 3-4 моделі OSI були

найпопулярнішою ціллю у кварталі, що становлять 65,5% від усіх спостережуваних атак[11].

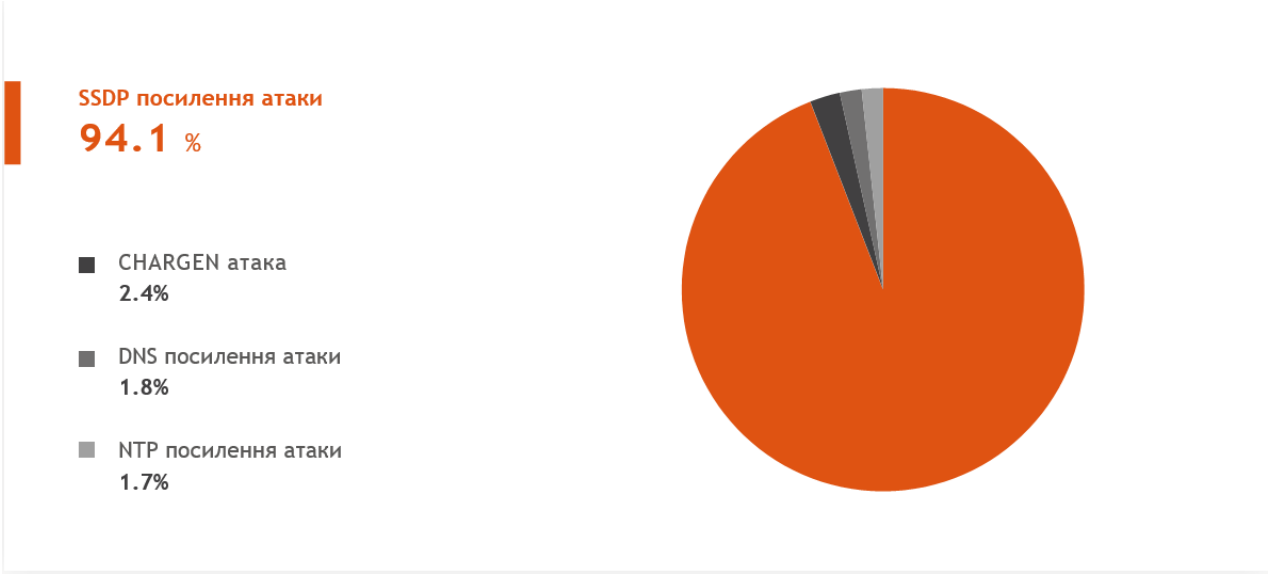

Рисунок 2.2 – Методи посилення DDoS атак[11].

Висновки до розділу 2

В даному розділі був розглянутий стан використання вразливості SSDP протоколу на основі звітів лідерів світового ринку по захисту від DDoS атак.

Даний розділ дає нам чітке представлення про те на скільки розповсюдженою являється вразливість SSDP протоколу і скільки при цьому проміжних хостів може бути скомпрометовано у зв’язку з її використанням. Хоча й вразливість даного протоколу не є абсолютно новою, проте різкий зріст використання цього вектору атаки свідчить про те, що він почав особливо часто використовуватись в останні роки і являється надзвичайно актуальним.

3 ПРОВЕДЕННЯ АТАКИ В ЛАБОРАТОРНИХ УМОВАХ ТА ВПРОВАДЖЕННЯ ЗАХИСТУ ВІД АТАКИ

3.1 Реалізація атаки

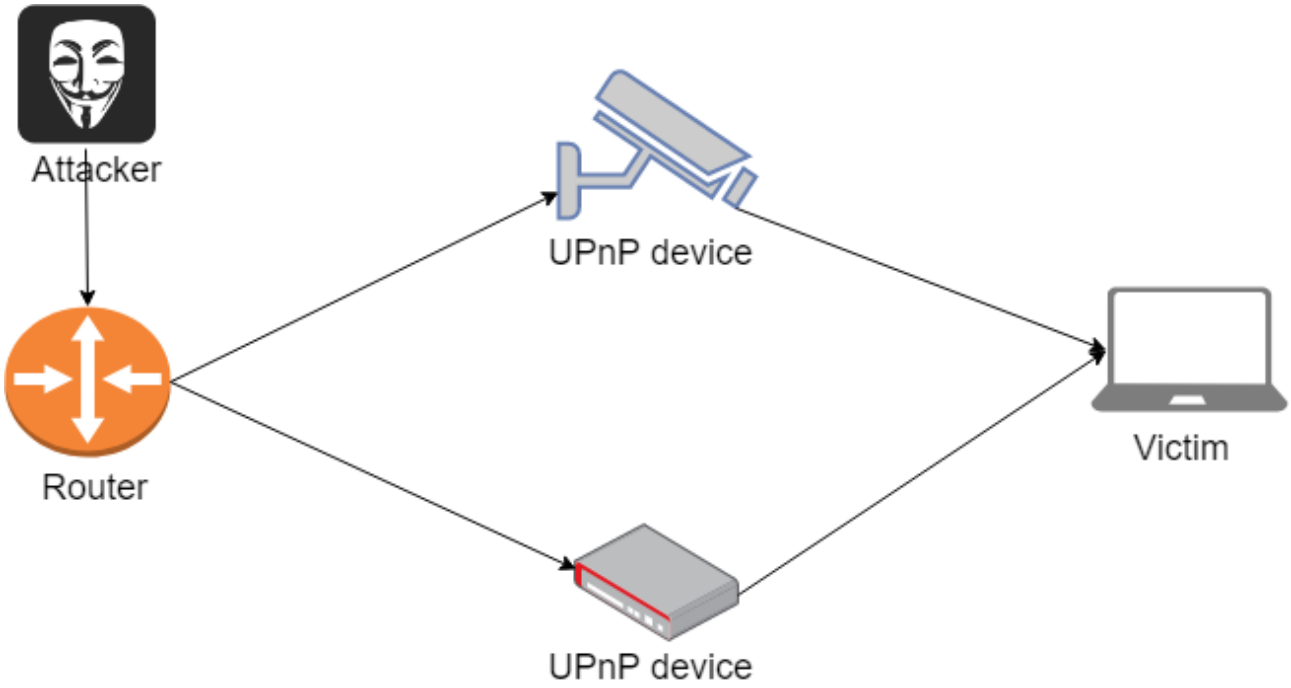

Рисунок 3.1 – UPnP атака

Для реалізації нашої атаки я використав вразливість набору протоколів UPnP, а саме SSDP. Некоректні базові налаштування пристроїв дають змогу звертатись до них із глобальної мережі. При відсутності механізму автентифікації, ми можемо скористатись цією вразливістю та підвищити об’єм пакетів, які будуть надсилатись на адресу жертви.

Формуємо пакети для відправлення за допомогою бібліотеки scapy, яка надає можливості створювати пакети. За допомогою меседжу M-Search ідентифікуємо вразливі пристрої. Код на рисунку 3.2 корисно виконати для пошуку вразливостей у власній локальній мережі, адже цілком можливо, що і у вас може бути незахищений UPnP пристрій.

```
from scapy.all import *

dest = "93.100.235.198"
source = "192.168.43.6"

msg = \
    'M-SEARCH * HTTP/1.1\r\n' \
    'HOST:239.255.255.250:1900\r\n' \
    'ST:ssdp:allp\r\n' \
    'MX:2\r\n' \
    'MAN:"ssdp:all"\r\n' \
    '\r\n'

p = IP(dst=dest, src=source) / UDP(sport=1900, dport=1900) / Raw(load=msg)

print srloop(p)
```

Рисунок 3.2 – Код для пошуку вразливих пристроїв

Список вразливих пристроїв можна також подивитись на сайті shodan. Список не є повністю актуальним та оновлюється з великою затримкою, але приблизну ситуацію описує. За допомогою пошуку rootDesc.xml, наявність якого є достатньою умовою вразливості UPnP пристрою.

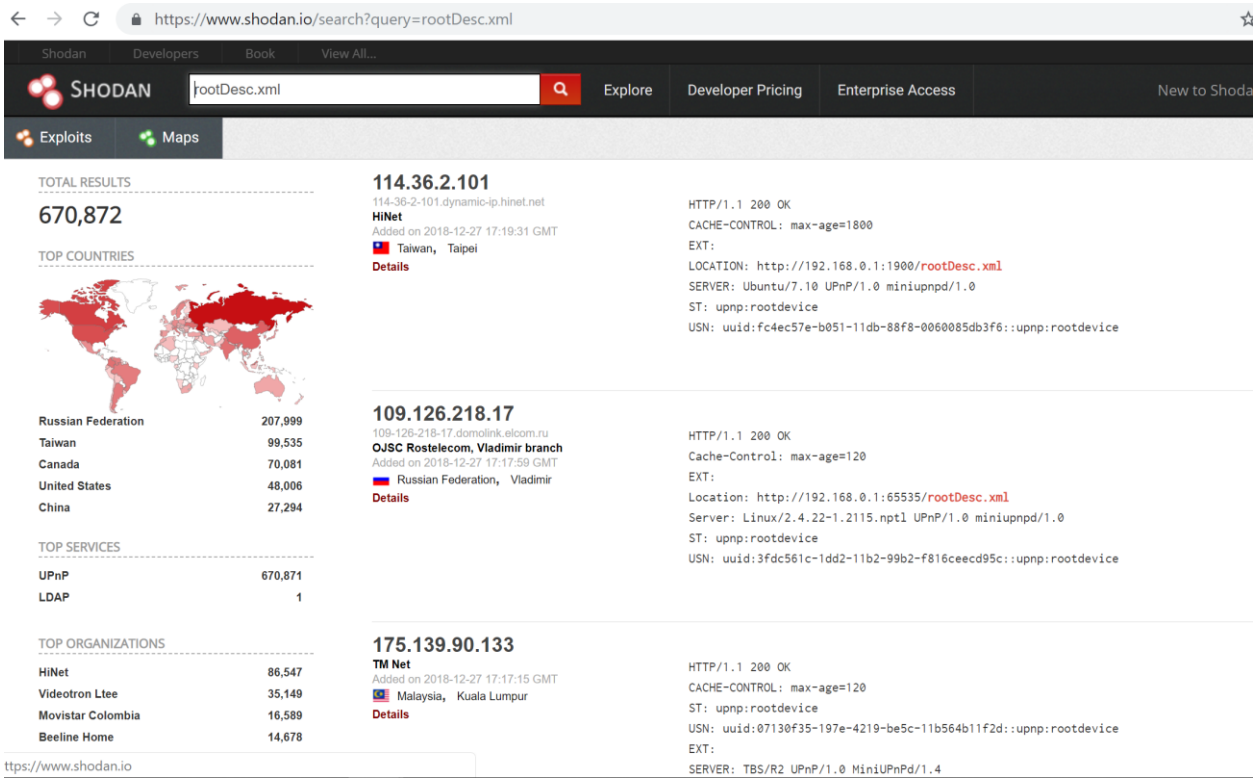

Рисунок 3.3 – Список вразливих пристроїв із сайту shodan

В результаті виконання даного коду ми надіслали пакет запит та отримали відповідь від вразливого пристрою, запит зображено на рисунку 3.4. У пакеті відповіді ми отримали HTTP-місцезнаходження файлу опису пристрою.

Рисунок 3.4 – Запит на UPnP пристрій

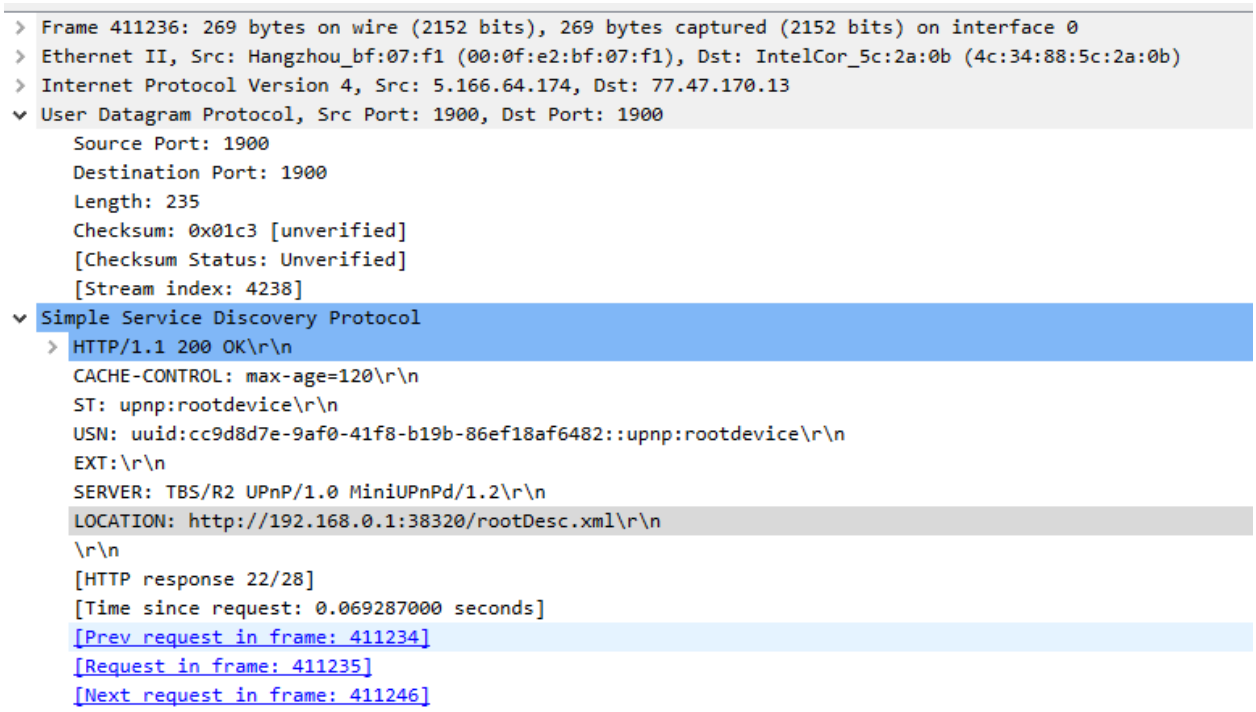

Рисунок 3.5 - Відповідь від UPnP пристрою

У коді можна додатково вказати множину IP-адрес, які ви забажаєте просканувати. Зібравши список вразливих пристроїв, зловмисник може надіслати запити на вразливі пристрої і отримати посилену відповідь на адресу жертви. Розмір відповіді і коефіцієнт посилення може змінюватися в залежності від вмісту файлу опису пристрою.

```
from scapy.all import *

dest = "93.100.235.198"
#source = "77.47.170.13"
source = "192.168.0.104"

msg = \
    'M-SEARCH * HTTP/1.1\r\n' \
    'HOST:239.255.255.250:1900\r\n' \
    'ST:ssdp:all\r\n' \
    'MX:10\r\n' \
    'MAN:"ssdp:all"\r\n' \
    '\r\n'

p = IP(dst=dest, src=source) / UDP(sport=1900, dport=1900) / Raw(load=msg)

send(p, loop=1)
```

Рисунок 3.6 – Код для використання вразливості SSDP

За допомогою даного коду ми підмінюємо IP-адресу джерела на адресу цілі атаки. Таким чином відповіді від пристрою будуть надходити на IP-адресу жертви. Дана програма буде створювати корисне навантаження до тих пір поки ми власноруч не зупинимо процес. При цьому кожний запит в свою чергу заставлятиме генерувати підсилену відповідь у адресу жертви. За допомогою нескладних маніпуляцій список вразливих пристроїв може бути розширений за рахунок сканування мережі. При цьому всьому, аналізуючи пакети, які надійшли на адресу жертви ми не зможемо дізнатися адресу відправника.

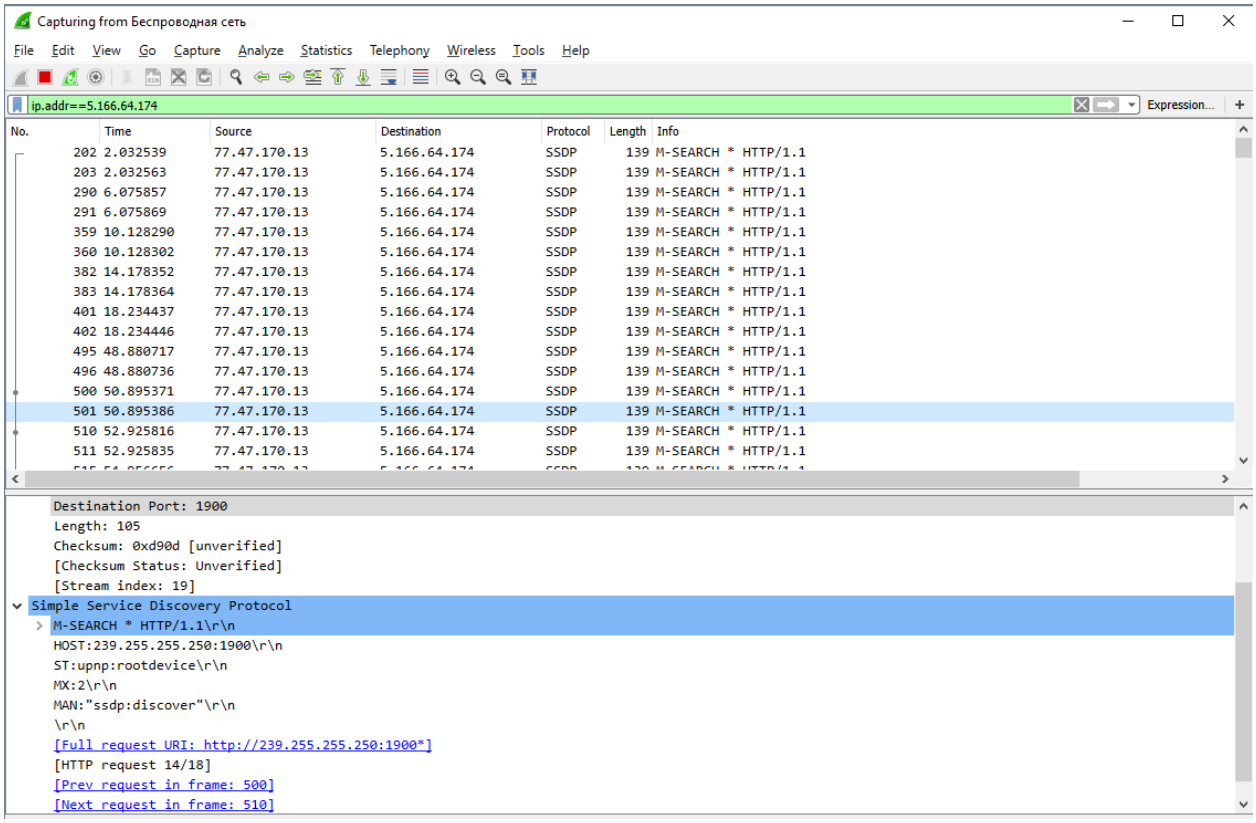

Рисунок 3.7 – Запити, надіслані із комп’ютера зловмисника

На рисунку 3.7 ми бачимо, що зловмисник надсилає запити на виявлення із адреси 77.47.170.13 на адресу 5.166.64.174. Але оскільки пакети є підробленими за допомогою спеціального бібліотеки, то відповіді на наші запити надходитимуть на іншу адресу.

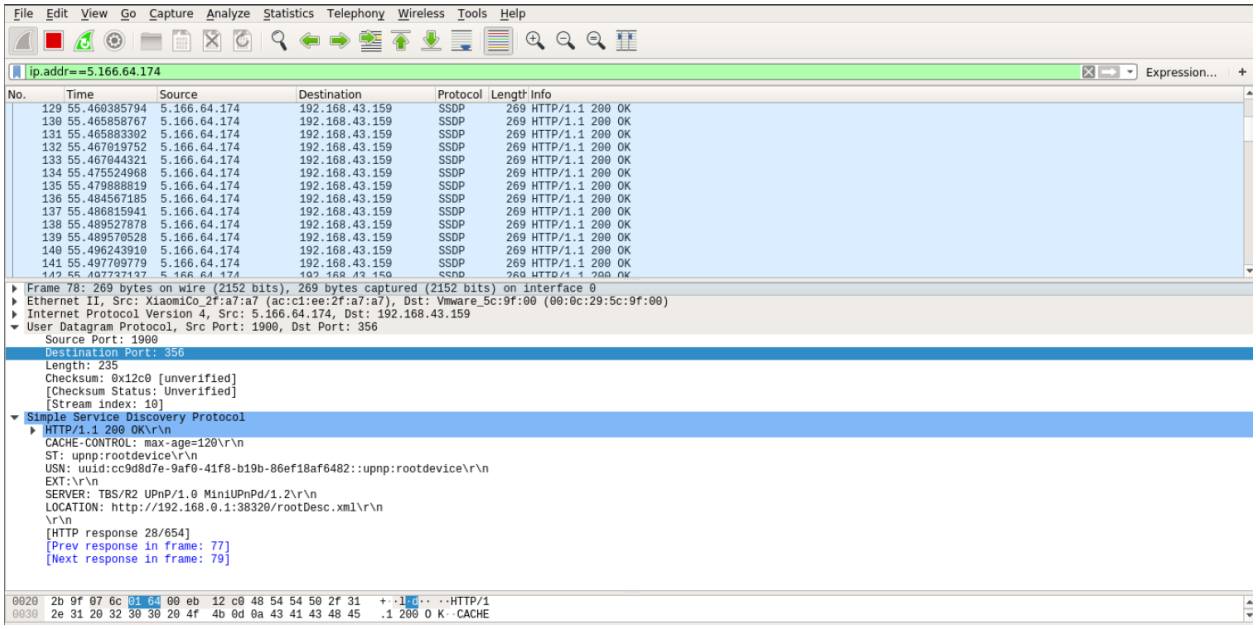

Рисунок 3.8 – Відповіді від UPnP пристрою

На рисунку 3.8 ми можемо побачити, що джерелом відповіді є адреса 5.166.64.174, а адресою призначення є 192.168.43.159, що свідчить про те, що підміна адреси була успішною. Також у заголовку LOCATION знаходиться унікальна адреса XML файлу з налаштуваннями UPnP пристрою.

Під час проведення досліду було надіслано близько 40 тис. пакетів та отримано близько 300 тис. пакетів. Розмір пакету запиту є статичним та складає 128 байт. Середній розмір пакету відповіді складає 342 байти. Таким чином за допомогою даної вразливості у лабораторних умовах рівень посилення атаки склав 20 разів.

3.2 Захист за допомогою Cisco Firepower NG Firewall

Рішення Cisco Firepower Management Center - це адміністративний центр для цілого ряду продуктів безпеки Cisco, які виконуються на різних платформах. Рішення забезпечує повне, уніфіковане управління міжмережними екранами, контролем додатками, запобіганням вторгнень, фільтрацією URL-адрес і захистом від удосконаленого шкідливого програмного забезпечення. Центр

управління - це центральна точка для управління подіями і політиками. Рішення Cisco Firepower Management Center дозволяє отримати детальну аналітичну інформацію про користувачів, додатки, пристрої, загрози і вразливості, існуючі у вашій мережі . Це рішення використовує цю інформацію для аналізу вразливостей вашій мережі, а потім надає адаптовані рекомендації про те, які політики безпеки слід впровадити, а які події безпеки вивчити і розслідувати. Центр управління надає зручні у використанні екрани політик для контролю доступу та захисту від відомих атак. Рішення інтегрується з захистом від удосконаленого шкідливого програмного забезпечення і «пісочницею», а також надає інструменти для відстеження інфікованого шкідливого програмного забезпечення по всій вашій мережі.

Дана система мережного захисту дає можливість гнучко налаштовувати аналіз трафіку, який проходить через сенсор Firepower Threat Defence. За допомогою даної системи ми маємо можливість налаштувати захист на двох різних етапах, а саме в acces control policy та в prefilter.

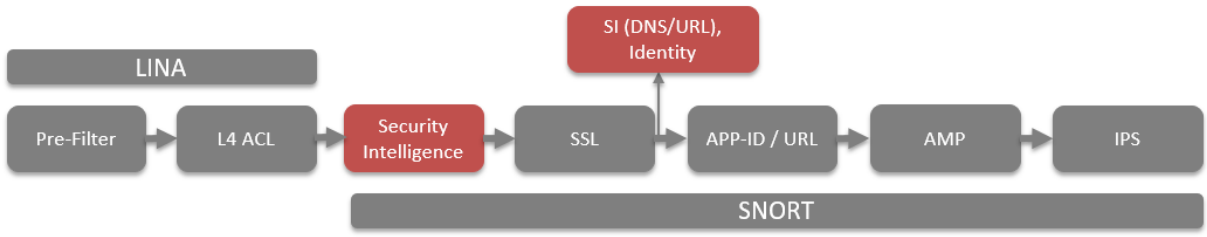

Рис 3.10 – Проходження пакетів у Cisco Firepower NG Firewall

На рисунку 3.10 зображено потоки данних, які обробляються в Cisco Firepower NG Firewall. З цього рисунку ми можемо зробити висновки, що для того, щоб не навантажувати систему додатковим трафіком, який проходитиме різні етапи дослідження трафіку, ми маємо змогу заборонити доступ за допомогою налаштування префільтру. Префільтр працює до 4 рівня по семирівневій моделі OSI. Оскільки дана атака використовує UDP та 1900 порт, то потрібно просто

створити відповідне правило, яке забороняло б проходження UDP трафіку через 1900 порт.

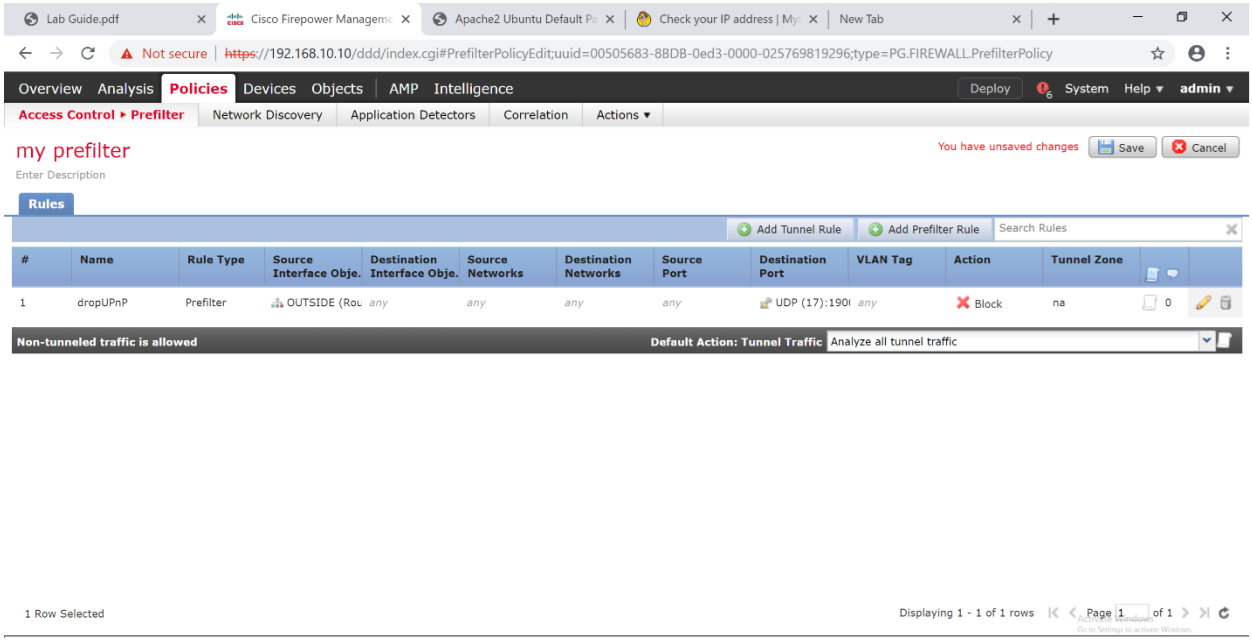

Рис 3.11 – Сформоване правило для префільтру

Після впровадження цього правила ми знову запустили програму для ініації SSDP атаки, але у вкладці аналіз не було відповідних записів про надходження трафіку з використанням SSDP. Очевидно, wireshark також не перехватив SSDP пакетів, тому що пакети відкидались на рівні префільтр і комп'ютеру не доводилось обробляти отримані пакети. Таким чином ми повністю справились з даним типом загрози за допомогою Cisco Firepower NG Firewall, як і очікувалось. Але, на жаль, пом'якшити саму атаку жертва не може, тому що атака направлена на те, щоб переповнити трафік жертви об'ємом данних, що в свою чергу буде викликати значне погіршення швидкості передачі трафіку у сторону жертви.

3.3 Рекомендації щодо уникнення схожих вразливостей

Список рекомендацій, які варто, на мою думку, втілити, як результат розбору вразливості SSDP протоколу:

- Постачальники Інтернет-послуг ніколи не повинні дозволяти виконувати підміну IP-адрес у їх мережі. Підміна IP є справжньою першопричиною проблеми.
- Інтернет-провайдери повинні дозволити своїм клієнтам використовувати BGP flowspec для обмеження вхідного трафіку, щоб зменшити перевантаження під час великих DDoS-атак.
- Розробники повинні думати про безпеку при створенні нових додатків і протоколів. UDP слід уникати, якщо не потрібна низька затримка, і у випадку якщо використовується UDP, протокол повинен мати певну форму автентифікації і ніколи не повинен дозволяти більше однієї відповіді на запит. Значення всіх відповідей має бути меншим або рівним запиту, які генерують їх.
- Інтернет-провайдери повинні збирати зразки потоку данних. Ці данні необхідні для визначення справжнього джерела атаки. Із збереженого потоку данних провайдер зможе визначити, який з клієнтів надсилав шкідливий трафік. Для відстеження DDoS-атак достатньо зберігати 1 пакет із 100000, при цьому зберігається конфіденційність окремих підключень клієнтів.
- Адміністратори повинні правильно налаштовувати міжмережний екран і дозволяти доступ до послуг для тих, хто їх потребує, а не для всього Інтернету. Деякі типи відповідей можуть бути заблоковані в межах програми або на рівні міжмережного екрану.

Що стосується UPnP:

- Не довіряти вхідним данним.
- Розглядати LAN як WAN.
- Робити тестові реалізації та активно слідкувати за вразливостями.
- Залучати до розробки людей з профільними знаннями із безпеки.
- Реалізувати аутентифікацію.
- Перейти на протокол HTTPS.

- M-SEARCH має практичний сенс як багатоадресний запит лише в локальній мережі.
- Підтримка одноадресного надсилання M-SEARCH повинна бути заборонена або принаймні обмежена в кількості відповідей в секунду, так само як реалізовано в DNS серверах.
- Відповіді M-SEARCH повинні надсилатись лише до локальної мережі. Відповіді, передані в мережу Інтернет, не мають сенсу і відкривають описану вразливість в дипломній роботі, одним із очевидних рішень даної проблеми є зміна TTL по замовчуванню на 1 або 2.

Висновки до розділу 3

В даному розділі був запропонований метод реалізації атаки на відмову в обслуговуванні за допомогою спеціально написаного коду із використанням бібліотеки scapy на мові python. Також в цьому розділі наведена покрокова інструкція для перевірки на наявність вразливих пристроїв у власній мережі, а також реалізації атаки в лабораторних умовах. Результатом наукового дослідження в даному розділі є метод захисту від вразливості SSDP протоколу.

Метод захисту реалізований в даному розділі може використовуватись у системах з аналогічним функціоналом та не є унікальним рішенням лише для даної системи. Також у даному розділі був сформований список рекомендацій для розробників протоколів UPnP та основні проблеми, які стали наслідком виникнення описаної вразливості.

ВИСНОВКИ

Результатом даної роботи є метод захисту UPnP пристроїв від вразливості SSDP протоколу. За допомогою методу захисту ми нейтралізуємо вразливість, яка дає змогу посилювати об’єм трафіку зловмисника у 20 разів і при цьому ускладнює, фактично унеможливлює, відслідковування реального джерела розподіленої атаки на відмову в обслуговуванні. Проте вразливість все ще залишається і даний метод захисту не являється рішенням проблеми. Розробники протоколу повинні реалізувати повноцінну роботу над помилками допущеними під час реалізації даного протоколу для того, щоб не просто нейтралізувати вразливість, а для того щоб реалізація протоколу не допускала використання SSDP у посиленні DDoS атак.

Результатом даної роботи також є список рекомендацій для подальшої реалізації у наборі UPnP протоколів, оскільки даний проект є закритим. Також даний метод захисту був реалізований на підприємстві під час проходження. Окрім нейтралізації вразливості описаної в даній роботі, ми також за допомогою методу захисту значно знизимо навантаження на інтегральну схему специфічного застосування у Cisco Firepower Threat Defence, якій би довелось аналізувати відповідний трафік та додатково навантажувати систему під час DDoS атаки з використанням даної вразливості.

Побічним результатом даної роботи є виявлення критичної вразливості в роботі Інтернет провайдерів, а саме можливість підміни IP адрес. Саме можливість підміни IP адреси являється причиною реалізації будь-якої атаки з використанням посилення атаки.

Наприклад, DNS протокол дозволяє посилювати атаку в 70 разів, що надзвичайно збільшує можливості зловмисників[12]. Отже, зловмисникам на сьогоднішній день не потрібно навіть мати великої пропускної здатності для реалізації потужних DDoS атак. Як наслідок на сьогоднішній день можлива реалізація атак з потужністю 1,3 Тбіт/сек і посиленням у 50000 разів[13].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. "DDoS, Machine Learning, Measures". // "Understanding Denial-of-Service Attacks". / , 2016. – (Taylor & Francis Group). – (ISBN:13: 978-1-4987-2965-9). – С. 12–34.

2. The Crossfire Attack. // IEEE Symposium on Security and Privacy. – 2013. – С. 127–142.

3. The Coremelt Attack [Електронний ресурс] – Режим доступу до ресурсу: https://netsec.ethz.ch/publications/papers/studer_esorics09.pdf.

4. D-FACE: An anomaly based distributed approach for early detection of DDoS attacks and flash events. // Journal of Network and Computer Applications. – 2018. – С. 49–63.

5. UPnP Design by Example, 2003. – (Intel Press). – (ISBN 0-9717861-1-9).

6. Internet Printing Protocol/1.0: Encoding and Transport [Електронний ресурс]. – 1999. – Режим доступу до ресурсу: https://tools.ietf.org/html/rfc2565.

7. Security Flaws in Universal Plug and Play. // Rapid7. – 2013. – С. 10–11.

8. How DHCP Works [Електронний ресурс] // Oracle Corporation. – 2010. – Режим доступу до ресурсу: https://docs.oracle.com/cd/E19455-01/806-0916/6ja8539a4/index.html.

9. New DDoS Attack Method Demands a Fresh Approach to Amplification Assault Mitigation [Електронний ресурс]. – 2018. – Режим доступу до ресурсу: https://www.imperva.com/blog/new-ddos-attack-method-demands-a-fresh-approach-to-amplification-assault-mitigation/.

10. SSDP reflection DDoS attacks [Електронний ресурс] // Akamai. – 2014. – Режим доступу до ресурсу: https://www.akamai.com/fr/fr/multimedia/documents/state-of-the-internet/ssdp-reflection-ddos-attacks-threat-advisory.pdf.

11. Threat Report DDoS. // NexusGuard. – 2018. – №3. – С. 7–8.

12. What is a DNS amplification attack [Електронний ресурс] – Режим доступу до ресурсу: https://www.imperva.com/learn/application-security/dns-amplification/.

13. Famous DDoS Attacks | The Largest DDoS Attacks Of All Time [Електронний ресурс] – Режим доступу до ресурсу: https://www.cloudflare.com/learning/ddos/famous-ddos-attacks/.