DOI 10.15276/imms.v14.no1-2.70
УДК 004.85

Informatics and Mathematical Methods in Simulation
Vol.14 (2024), No. 1-2, pp.70-84

ЗАСТОСУВАННЯ МЕТОДУ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ АНАЛІЗУ ШКІДЛИВОГО МЕРЕЖЕВОГО ТРАФІКУ НА КАНАЛЬНОМУ РІВНІ (ARP-атаки)

В.В. Палагін[1], О.А. Палагіна[2], О.В. Івченко[3], О.М. Панаско[4], Р.Л. Пташкін[5]

[1-4]Черкаський державний технологічний університет
460, Шевченка б-р, м.Черкаси, 18006, Україна
[5]Черкаський науково-дослідний експертно-криміналістичний центр МВС України,
104, Пастерівська вул., м.Черкаси, 18009, Україна
emails: palahin@ukr.net[1], palahina@ukr.net[2],
sania_ivchenko@ukr.net[3] , lena.pa@ukr.net[4], ndekc.ck@gmail.com[5]

Широке розповсюдження програмно-визначених мереж (*Software-Defined Networking* - SDN), *Internet of Things* (IoT) мереж забезпечило неперевершену гнучкість і ефективність керування мережами, але водночас поставило нові виклики у захисті мережевої інфраструктури. Однією з важливих загроз залишаються атаки підробки протоколу розпізнавання адрес (*Address Resolution Protocol* - ARP), що порушують цілісність мережі та конфіденційність даних. У цьому рукописі представлено новий підхід до виявлення ARP-спуфінгу в мережах при аналізі обмежень існуючих методологій. Проведено аналіз ARP протоколів, їх призначення та основних методів захисту від атак. Наведено типові загрози комп'ютерним мережам фізичного та канального рівнів моделі OSI та проведено аналіз особливостей виявлення таких загроз з використанням методів штучного інтелекту – ШІ (*Artificial intelligence* - AI). Запропоновано застосування методів машинного навчання (*Machine learning* - ML) для аналізу трафіку на основі отримання даних в режимі реального часу з платформи Wireshark. Новий метод базується на використанні ШІ для класифікації та виявлення зловмисного мережевого трафіку, згенерованого в результаті атак, що використовують протокол ARP. Розроблена модель та метод демонструють виняткову надійність, досягаючи 100% точності виявлення ARP-спуфінгу, що має вирішальне значення для підтримки швидкості реагування мережі. Результати аналізу можуть бути використані для прийняття обґрунтованих рішень щодо вибору методів захисту для мереж різного призначення та з різними вимогами щодо захисту інформації. Використання ШІ для моніторингу та аналізу мережевого трафіку дозволяє значно підвищити ефективність і швидкість виявлення загроз. Завдяки здатності адаптуватися до нових типів атак і виявляти складніші шаблони аномалій, запропонований підхід забезпечує більш високий рівень безпеки мережевої інфраструктури. Це дослідження демонструє потенціал інноваційних технологій у боротьбі з кіберзагрозами та сприяє розвитку надійних методів захисту для сучасних мереж.

Ключові слова: аналіз мережевого трафіку, ARP Spoofing, штучний інтелект, атаки на рівні L2

Вступ. Виявлення загроз типу ARP-атак залишається надзвичайно актуальним у сучасному мережевому середовищі. Незважаючи на розвиток технологій безпеки, ARP-атаки продовжують бути ефективними засобами для порушення конфіденційності, цілісності та доступності даних у мережах. Це пояснюється тим, що ARP-протокол, який був розроблений багато років тому, не має вбудованих механізмів захисту, і тому залишається вразливим до підробки та інших форм експлуатації.

Розповсюдження IoT-пристроїв та SDN мереж додатково збільшує поверхню для атак. IoT-пристрої часто мають обмежені можливості для забезпечення безпеки і можуть стати легкою мішенню для зловмисників, які здійснюють ARP-спуфінг. Також

SDN, з його централізованим управлінням мережею, може бути серйозно уражений, якщо атака успішно проведена, що може призвести до значних порушень у мережевій інфраструктурі.

Незважаючи на те, що архітектура SDN виступає як надійна структура, що підвищує безпеку мережі та оптимізує мережеве адміністрування, вона не усуває різні форми атак підробки. Серед них атаки *Distributed Denial of Service* (DDoS) і *Man-in-the-Middle* (MitM) залишаються серйозними загрозами, потенційно скомпрометувавши конфіденційні дані користувачів у мережі. Примітно, що одним із найпоширеніших вторгнень у локальні мережі (LAN) є підробка ARP.

Актуальність виявлення ARP-атак також підсилюється зростаючими вимогами до кібербезпеки у різних секторах, таких як фінанси, охорона здоров'я, промисловість та державні установи. У цих сферах порушення мережевої безпеки може мати серйозні наслідки, включаючи фінансові втрати, втрату конфіденційної інформації та підрив довіри до організації.

Пристрої мережі, які працюють на другому рівні еталонної моделі OSI (*Open Systems Interconnection*) вважаються найслабшою ланкою в інфраструктурі безпеки [1-3, 6]. Розповсюджена IT-політика BYOD (*Bring Your Own Device*), використання віртуальних мереж SDN і низки складних атак, збільшують вірогідність того, що мережі стають більш уразливими до проникнення саме на рівні L2. Протоколи рівня L2 дуже часто залишаються без належної уваги і здебільшого працюють зі стандартною конфігурацією. Слід пам’ятати, що порушення мережної безпеки на рівні L2 також впливатиме на всі рівні, розташовані вище. Таким чином, фахівцям з мережної безпеки потрібно також запобігати і вчасно нейтралізувати атаки на інфраструктуру LAN рівня L2.

Поширена загроза підробки ARP створює значний ризик для безпеки комп’ютерних мереж, що призводить до потенційного підслуховування, фальсифікації та порушення мережевого трафіку. Виявлення ARP-атак є складним завданням, і їх наслідки можуть бути серйозними, включаючи крадіжку даних і вразливість мережі. Ця вразливість у поєднанні з недоліками протоколів ARP відкриває зловмисникам шляхи для використання даних топології мережі, що призводить до різних провалів безпеки.

В роботі проведений аналіз атак, які базуються на маніпулюванні протоколом ARP. Наведено приклади трафіку, який утворюється при активізації атаки з використанням протоколу ARP. Використання методів машинного навчання для виявлення ARP-атак представляє собою інноваційний підхід, який дозволяє підвищити точність і ефективність захисту мереж. Завдяки здатності ML-алгоритмів аналізувати великі обсяги даних у режимі реального часу і виявляти аномалії, сучасні системи захисту можуть більш ефективно ідентифікувати та реагувати на загрози, забезпечуючи більш високий рівень безпеки для мереж різного масштабу і призначення.

Мета даної роботи полягає у підвищенні безпеки виявлення ARP-атак на основі використання методів штучного інтелекту для аналізу мережевого трафіку в режимі реального часу, що включає детальний аналіз ARP протоколів та існуючих методів захисту, оцінку їх обмежень, а також застосування інструментів для збору та аналізу даних, таких як Wireshark. Основною задачею є підвищення точності та швидкості виявлення ARP-спуфінгу, забезпечуючи надійний захист мережевої інфраструктури в умовах зростаючих кіберзагроз, зокрема у середовищах з великою кількістю IoT-пристроїв та SDN мереж.

1. Аналіз та особливості атак на протоколи ARP. ARP – це широко використовуваний протокол, який перетворює IP-адресу на MAC-адресу (*Media Access Control*). В більшості випадків ARP необхідний для того, щоб пристрої могли знаходити один одного в межах одного сегменту мережі. Цей протокол застосовується при організації мереж за протоколом TCP/IP (з використанням протоколу IPv4) для перетворення IP-адреси в MAC-адресу і визначений у стандарті RFC 826. У процесі

перетворення адрес за протоколом ARP застосовуються лише два типи пакетів: ARP-запит та ARP-відповідь. Трафік з використанням протоколу ARP зазвичай виникає в процесі обміну даними по мережі тоді, коли MAC-адреса одержувача невідома. Пристрій, що передає, спочатку шукає цю адресу у своєму кеші. Якщо адреса відсутня в кеші, то вона може бути отримана шляхом додаткового обміну даними по мережі з використанням протоколу ARP. Як тільки процес перетворення адрес завершиться, передавальний пристрій оновить свою кеш-пам'ять і помістить у неї відповідність MAC і IP-адрес приймального пристрою і почне передачу даних. Приклад ARP-таблиці хосту (отримана командою *arp –a*) наведена на рис. 1.

```
PS C:\Users\robst> arp -a

Interface: 192.168.1.246 --- 0x3
  Internet Address      Physical Address      Type
  192.168.1.1           bc-76-c5-1d-19-56     dynamic
  192.168.1.245         f8-ac-65-86-fb-28     dynamic
  192.168.1.255         ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

Рис. 1. Приклад ARP-таблиці хосту

Для аналізу мережевого трафіку в режимі реального часу використовувалася платформа *Wireshark* [6], яка є потужним інструментом для виявлення, діагностики та усунення проблем у мережах. Основні призначення *Wireshark* включають захоплення пакетів даних, які проходять через мережу, надання детальної інформації про кожен пакет, діагностування проблем з підключенням, затримками, втратами пакетів, виявлення аномалій та ін. *Wireshark* використовується для виявлення потенційних загроз та аномальної активності в мережі, таких як ARP-атаки, DoS-атаки та інші види зловмисного трафіку. Приклад перехвату типового трафіку між пристроями інструментом *Wireshark* з використанням протоколу ARP наведений на рис. 2.

arp

No.	Time	Source	Destination	Protocol	Length	Info
81…	-35.757171	XiaomiMobile_8d:13:f2	Broadcast	ARP	60	Who has 192.168.31.251? Tell 192.168.31.1
82…	-35.747205	XiaomiMobile_8d:13:f2	Broadcast	ARP	60	Who has 192.168.31.252? Tell 192.168.31.1
83…	-35.737161	XiaomiMobile_8d:13:f2	Broadcast	ARP	60	Who has 192.168.31.253? Tell 192.168.31.1
84…	-35.727217	XiaomiMobile_8d:13:f2	Broadcast	ARP	60	Who has 192.168.31.254? Tell 192.168.31.1
47…	5.953990	XiaomiMobile_8d:13:f2	HewlettPacka_0a:3e:d1	ARP	60	Who has 192.168.31.150? Tell 192.168.31.1
47…	5.954007	HewlettPacka_0a:3e:d1	XiaomiMobile_8d:13:f2	ARP	42	192.168.31.150 is at 84:a9:3e:0a:3e:d1

Рис. 2. Типовий ARP трафік в середовищі Wireshark

З рисунку 2 видно, що ARP-запит посилається в широкомовному режимі (*broadcast message*), а ARP- відповідь надсилається в одноадресному режимі (*unicast message*). Також можливі випадки утворення трафіку, який складається з так званих самочинних ARP-пакетів (ARP-пакети, які генеруються автоматизованими засобами або зловмисниками для порушення нормальної роботи мережі), що також є типовим видом трафіку. Поява самочинних ARP-пакетів можна виявити у ряді випадків:

- зміна IP-адреси пристрою призведе до появи самочинних пакетів;
- при запуску деяких операційних систем відбувається передача самочинних ARP-пакетів;
- в ряді систем самочинні ARP-пакети слугують для підтримки балансування навантаження.

Приклад перехвату трафіку з самочинним ARP-пакетом наведений на рис. 3. З рисунку видно, що самочинні ARP-пакети розсилаються в широкомовному режимі.

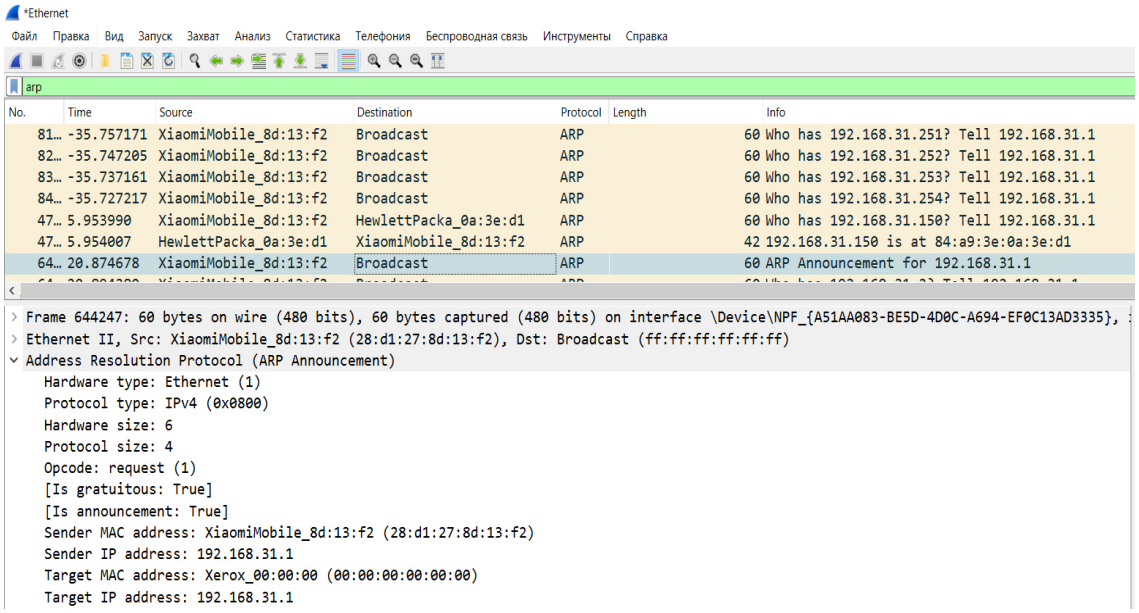

Рис.3. Типовий ARP трафік з самочинним ARP-пакетом

Протокол ARP був стандартизований багато років тому і ніколи не було способу гарантувати автентичність ARP-повідомлення. Як наслідок, є кілька атак ARP, які можуть неправильно спрямувати трафік в локальній мережі з метою його перехоплення в зловмисних цілях. Деякі з атак і методи, які використовуються для проникнення через ARP-протокол, включають:

1. ARP Спуфінг (ARP Spoofing) - є найбільш поширеним типом ARP-атак. Вона полягає у надсиланні фальшивих ARP-повідомлень в мережу для зміни асоціацій між IP-адресами та MAC-адресами. Це дозволяє зловмиснику перенаправляти трафік через свій пристрій. Основні види ARP-спуфінгу включають *Man-in-the-Middle (MITM)* атака та *Packet Sniffing*;

2. ARP Cache Poisoning - атаки передбачають надсилання фальшивих ARP-повідомлень для заповнення ARP-кешу мережевих пристроїв помилковими записами. В результаті цього пакети можуть бути спрямовані на неправильні або неіснуючі MAC-адреси, що призводить до порушення роботи мережі. Види ARP Cache Poisoning включають *Denial of Service (DoS* - перенаправлення трафіку на неіснуючі MAC-адреси, що призводить до неможливості досягти цільового пристрою) та *Network Disruption* (вплив на маршрутизацію трафіку, що призводить до дезорганізації роботи мережі);

3. ARP шторм (Flooding) - зловмисник надсилає велику кількість ARP-запитів або відповідей для перевантаження ARP-кешу мережевих пристроїв. Основні наслідки такої атаки є *Resource Exhaustion* (перевантаження кешу ARP може призвести до сповільнення роботи або навіть збою мережевих пристроїв) та *Network Congestion* (велика кількість ARP-пакетів може перевантажити мережу, зменшуючи її продуктивність);

4. ARP-пінг (ARP Ping) - це метод, який використовується для виявлення активних пристроїв у локальній мережі шляхом надсилання ARP-запитів. Зловмисники можуть використовувати ARP-пінг для збору інформації про пристрої в мережі перед проведенням інших атак.

Атака *ARP spoofing*, також відома як «*отруєння кешу*» *ARP*, використовується в атаці типу «людина посередині» [5, 6]. Під час атаки *ARP spoofing* зловмисник діє наступним чином: надсилає небажане, підроблене повідомлення відповіді *ARP*, яке містить підроблену MAC адресу машини зловмисника для всіх хостів у локальній мережі. Після отримання відповіді *ARP* усі пристрої в локальній мережі оновлять свої *ARP* або таблиці MAC-адрес із неправильною MAC-адресою. Це ефективно *«отруює кеш»* на кінцевих пристроях. Якщо таблиці *ARP* «*отруєні*», це дозволить зловмиснику

видати себе за інший хост, щоб отримати доступ до конфіденційної інформації. Таким чином, трафік спрямовується не на фактичний хост, а на хост із підробленою MAC-адресою. На наведеному нижче рисунку (рис.4) представлена атака ARP, де зловмисник надіслав фальшиву відповідь, яка «*отруїла кеш*» в пристроях. Усі хости в мережі тепер думають, що 10.40.10.103 знаходиться на 46:89:FF:4C:57, замість 00:80:68:B4:87.

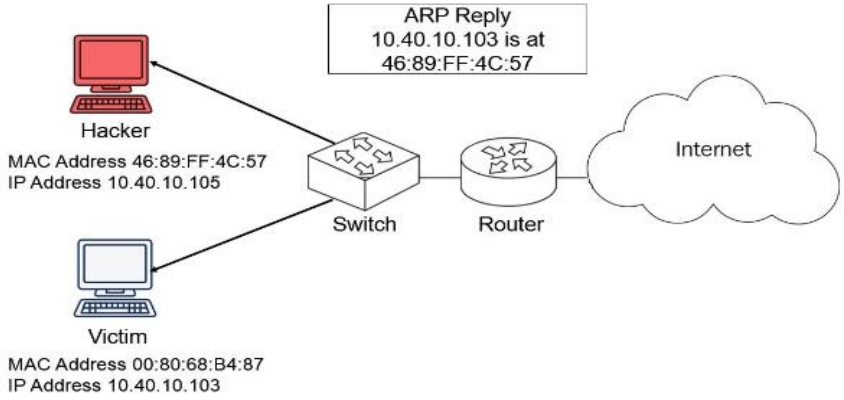

Рис. 4. Атака ARP spoofing

Зазвичай при проведенні атаки, націленої на отруєння кешу ARP, суб'єкт загрози може надсилати іншим хостам у підмережі *самочинні* ARP-відповіді, які містять MAC-адресу зловмисника і IP-адресу шлюзу за замовчуванням. Приклад трафіку, який можна спостерігати при атаці отруєння кешу ARP показано на рис. 5.

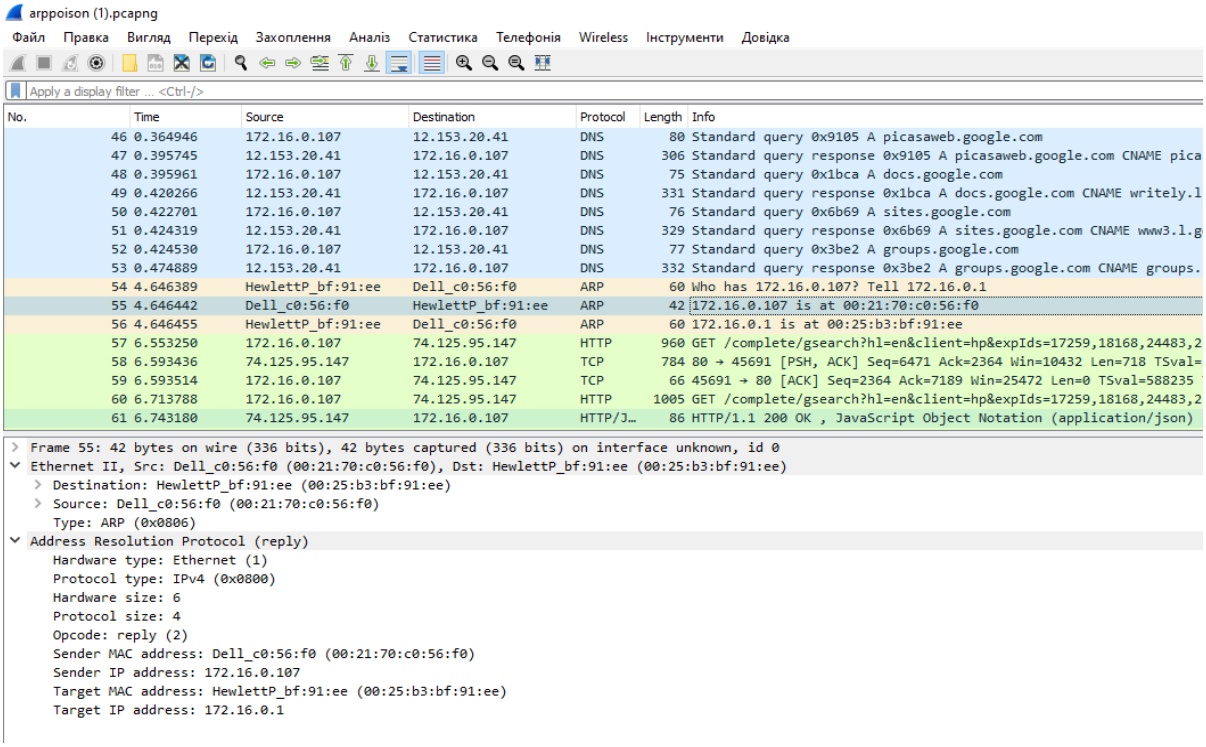

Рис. 5. Приклад трафіку при атаці типу отруєння кешу ARP

З наведеного прикладу видно, що хост зловмисника направляє одноадресний запит і одноадресну відповідь, в яких видає себе за шлюз з IP-адресою 172.16.0.1

Якщо зловмисник перенаправляє трафік, тоді він може перехопити його для отримання конфіденційної інформації, що може стати підготовкою до більш складної атаки.

Шторм ARP (Flooding). У локальній мережі типовим трафіком є ARP-повідомлення запитів/відповідей. При цій атаці можна спостерігати велику кількість ARP-запитів, тому ARP-шторм є формою атаки на відмову в обслуговуванні (DoS).

Для ефективної доставки даних комутатор використовує таблицю *Content Addressable Memory* (CAM), яка містить пари MAC-адрес і пов'язані з ними фізичні порти комутатора. Зловмисник може самовільно надсилати комутатору ARP-повідомлення із підробленою MAC-адресою, в результаті чого комутатор оновлюватиме свою MAC-таблицю. Оскільки всі MAC-таблиці мають обмежений розмір, комутатор може вичерпати ресурси для зберігання MAC-адрес. Атаки з переповнення таблиць MAC-адрес (MAC-флуд), користуючись цим обмеженням, бомбардують комутатор кадрами з підробленими MAC-адресами джерела, допоки таблиця MAC-адрес комутатора не заповниться.

Шторм ARP заповнює таблицю CAM комутатора і переповнює її тисячами фіктивних записів. У цей момент комутатор просто діє як концентратор і надсилає дані на всі порти. Наслідки такої атаки можна охарактеризувати наступним чином:

- у зловмисника з'являється можливість перехоплення всього трафіку з можливим наступним розкриттям конфіденційних даних;
- через велику кількість широкомовних запитів знижується пропускна здатність мережі.

Приклад трафіку, який можна спостерігати при атаці «ARP-шторм», показано на рис.6.

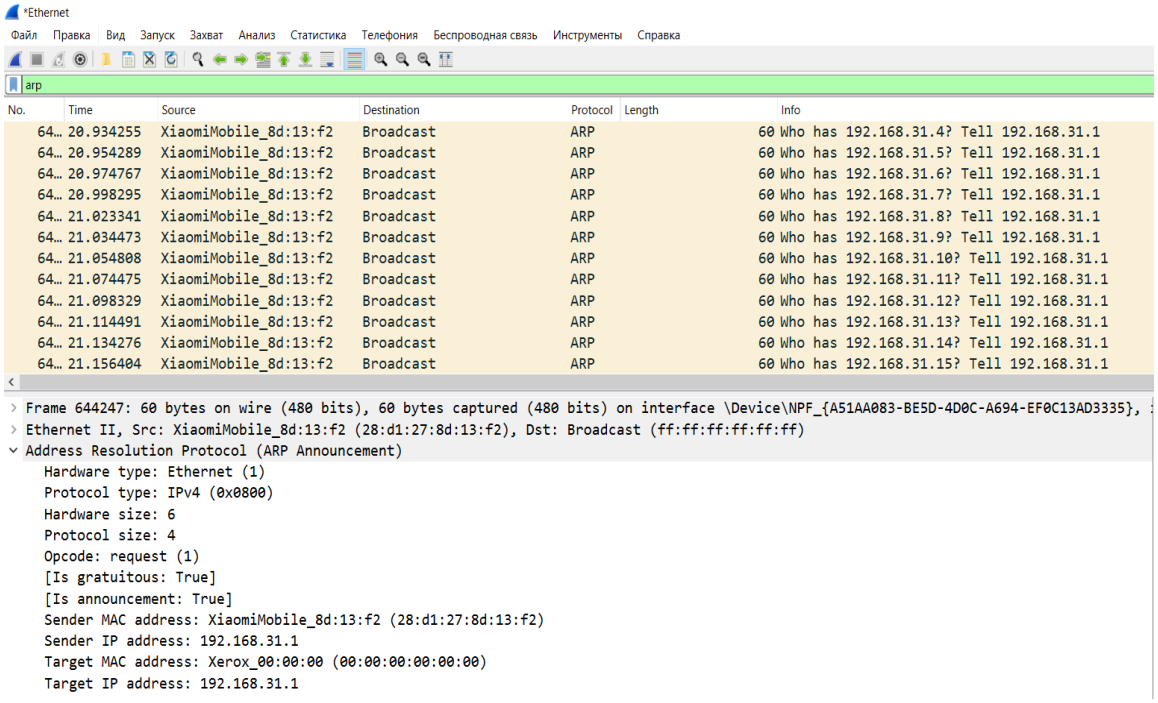

Рис.6. Приклад трафіку при атаці типу ARP-шторм

2. Методи захисту від ARP-атак. Виявлення трафіку, який спричинений атакою *ARP spoofing*, є досить складною операцією, так як потребує аналізу і виявлення шкідливого трафіку при активізації атаки *ARP spoofing* (початковий етап «*отруєння ARP кешу*») і аналізу і виявлення самого факту, що трафік переадресовується через вузол зловмисника. Щоб виявити цю загрозу використовують наступні підходи [5, 6]:

- системи виявлення/запобігання вторгненням (IDS/IPS - *Intrusion Detection Systems/Intrusion Protection Systems*), які налаштовують пристрої для відстеження аномальної активності ARP, наприклад штормів ARP, які зазвичай мають специфічні сигнатури. Пристрій має надіслати сповіщення, якщо буде виявлено небажані відповіді;
- статичні записи ARP - жорстке відображення адрес для запобігання спуфінгу. Це унеможливлює автоматичну актуалізацію ARP-кешу і захищає від підроблених

ARP-повідомлень. Однак цей метод вимагає адміністративних зусиль для підтримки і оновлення Це не найкращий метод, оскільки він погано масштабується у великих мережах;

- брандмауери – використовується лише список контролю доступу з фільтрацією пакетів авторизованого трафіку, який знаходиться в сегменті мережі;
- програмне забезпечення для захисту від ARP - це програмне забезпечення відстежує спуфінг, який може представлятися як дві IP-адреси з однаковою MAC-адресою, а також інші методи для виявлення зловмисної поведінки ARP.

Аналіз літературних джерел доводить, що на основі цих підходів проводяться багатогранні наукові дослідження. У роботі [7] продемонстровано методи щодо протидії атакам ARP-спуфінгу в межах SDN. Їх основною технікою є система виявлення та запобігання вторгненням (*Detection and Prevention System* - IDPS), що використовує технологію SDN. Цей IDPS динамічно адаптує параметри SDN для виявлення та запобігання підозрілим мережевим діям, додаючи до чорного списку шкідливі MAC-адреси. Для персоналізації та оцінки ефективності IDPS було розроблено спеціалізоване програмне забезпечення, інтегроване зі спеціальною бібліотекою для перевірки введених користувачем даних.

В роботі [8] проведене дослідження, спрямоване на боротьбу з обома основними типами ARP-атак у SDN. Запропоноване рішення розширює функціональні можливості контролера SDN шляхом включення спеціального модуля ARP. Цей модуль швидко виявляє та пом'якшує атаки, не перевантажуючи та не спричиняючи відмову в обслуговуванні (DoS) на контролері.

В роботі [9] представлена система, призначена для автономної ідентифікації та протидії мережевим вторгненням, з особливим акцентом на трафік ARP. Використовуючи підхід, заснований на складних обчисленнях, ця система визначає зловмисників або порушників і скасовує доступ до мережі, дозволяючи авторизованим користувачам продовжити роботу. Це підвищує продуктивність системи в таких сферах, як виявлення атак, пом'якшення та оптимізація пропускної здатності. Існують певні обмеження, наприклад, складність обчислень може вплинути на роботу у реальному часі.

В [10] запропоновано механізм, призначений для боротьби з підробкою ARP. Їхня система працює через спеціальну машину, яка співпрацює з контролером SDN для збору інформації про топологію мережі та ARP-запитів. Суть цього підходу полягає в перенаправленні ARP-трафіку на виділену машину, де спеціалізовані методи аналізують дані. Ці та інші методи передбачають наявність спеціального обладнання на базі контролерів SDN або окремих виділених машин, що не завжди може бути реалізованим, в тому числі при розгортанні мереж для пристроїв IoT або ресурсообмежених систем.

Впровадження сучасних методів машинного навчання (ML) для захисту від ARP-атак є актуальною та перспективною стратегією. Серед основних підходів щодо застосування ML відносяться наступні:

1. *Виявлення аномального ARP-трафіку* - моделі машинного навчання можуть бути навчені на основі великого обсягу даних ARP-трафіку для виявлення аномалій, наприклад:
   - *методи класифікації* - використання алгоритмів класифікації, таких як Random Forest, Support Vector Machines (SVM) або нейронні мережі для ідентифікації аномального ARP-трафіку порівняно з нормальним;
   - *кластеризація* - аналіз ARP-повідомлень для виявлення змін, які можуть свідчити про спроби підробки або інші аномалії.

2. *Застосування рішень з машинного навчання на мережевих пристроях* - деякі мережеві пристрої можуть підтримувати вбудовані моделі машинного навчання для моніторингу ARP-трафіку та автоматичного реагування на виявлені загрози, наприклад:

- *мережеві комутатори з DAI* - використання *Dynamic ARP Inspection* (DAI) для перевірки коректності ARP-повідомлень за допомогою навчених моделей;
- *системи виявлення вторгнень (IDS)* - інтеграція з системами IDS, які використовують машинне навчання для виявлення нестандартних патернів ARP-трафіку.

3. *Оновлення моделей і навчання в реальному часі* - враховуючи динамічну природу мережевих атак, важливо мати можливість постійно оновлювати моделі машинного навчання і навчати їх на нових даних. Це дозволяє підтримувати ефективність виявлення навіть у змінних умовах мережі.

Прикладом застосування ML для аналізу мережевого трафіку може бути робота [11], в якій пропонуються алгоритми для виявлення DDoS-атак у межах SDN. В цих алгоритмах використовуються алгоритм *K-means++*, доповнений алгоритмом *Fast k Nearest Neighbor*. Основою цього підходу є модульна система виявлення, яка повністю інтегрована в контролер SDN. Контролер періодично взаємодіє з комутаторами для оцінки та ідентифікації мережевих потоків. Якщо вхідний потік має ознаки DDoS-атаки, контролер негайно налаштовує правила пересилання таблиці потоків і надсилає сповіщення комутатору, організовуючи гнучку відповідь на аномалію. Через періодичну оцінку потоку можуть виникати потенційні накладні витрати на ресурси, що впливає на продуктивність мережі.

В роботі [12, 13] представлено метод штучного інтелекту на основі нейронних мереж для виявлення ARP-спуфінгу в мережах IoT. Запропоновані методи демонструють високу точність у виявленні ARP-спуфінгу в мережах Інтернету речей. Разом з цим необхідно відміти складність розгортання і навчання нейронних мереж, що потребує додаткових ресурсів на отримання моделі ML.

Детальний аналіз різноманітних підходів щодо виявлення ARP атак в мережах наведено в роботі [14], де представлені порівняльні характеристики ефективності різних методів.

3. Архітектура запропонованої моделі ML для аналізу ARP атак в реальному часі

Побудова простого методу машинного навчання для виявлення ARP-атак на локальному комп'ютері або мережі має свої переваги порівняно з застосуванням спеціалізованих контролерів SDN. До таких переваг можна віднести:

- *простота і швидкість впровадження* - простий метод машинного навчання може бути реалізований в короткі строки без значних інвестицій у нове обладнання або програмне забезпечення. Він може використовувати наявні дані і інфраструктуру, що значно спрощує процес впровадження і витрати на підтримку;
- *незалежність від інфраструктури* - метод ML може функціонувати навіть у стандартних мережевих середовищах без необхідності внесення значних змін у існуючу інфраструктуру. Це робить його універсальним і придатним для різних типів мереж та організацій;
- *гнучкість і адаптивність* - методи машинного навчання можуть бути легко адаптовані до нових атак і змін у мережевих умовах через оновлення моделей навчання «на льоту». Це дозволяє швидко реагувати на нові загрози і покращувати точність виявлення аномалій з часом;
- *ефективність в розподілених середовищах* - моделі машинного навчання можуть бути навчені локально на кожному комп'ютері або вузлі мережі, що дозволяє розподілене виявлення загроз без необхідності централізованого керування.

У той час, як спеціалізовані контролери SDN можуть надавати розширені можливості управління та моніторингу мережі, їх впровадження і підтримка можуть бути витратними і складними. Особливо це стосується малих і середніх підприємств, які можуть не мати необхідної інфраструктури або бюджету для впровадження SDN.

Отже, побудова простого та ефективного методу машинного навчання для виявлення ARP-атак може бути доступним рішенням для багатьох організацій,

забезпечуючи необхідний рівень безпеки мережі без значних інвестицій та складнощів, пов'язаних з SDN.

В роботі пропонуються спеціалізовані програмні рішення на мові Python, які базуються на застосуванні моделей машинного навчання, які своєчасно класифікують і виділяють ARP атаки та створюють керуючі сигнали для оповіщення. Розробка та застосування таких моделей *ШІ дозволяє автоматизувати аналіз* мережевого трафіка в реальному часі на основі даних з платформи *Wireshark* та підвищити безпеку комп'ютерних мереж.

На рис.7 наведена загальна структура підготовки даних для побудови моделі ML. Попередня обробка даних є важливою процедурою для забезпечення якості та надійності результатів аналізу даних і роботи моделей машинного навчання. Добре оброблені дані допомагають покращити точність та інтерпретованість моделей, а також зменшити можливість виникнення помилок під час аналізу. Для отримання моделі ML, яка буде завантажуватися і використовуватися для аналізу даних в реальному часі, потрібно виконати декілька підготовчих етапів.

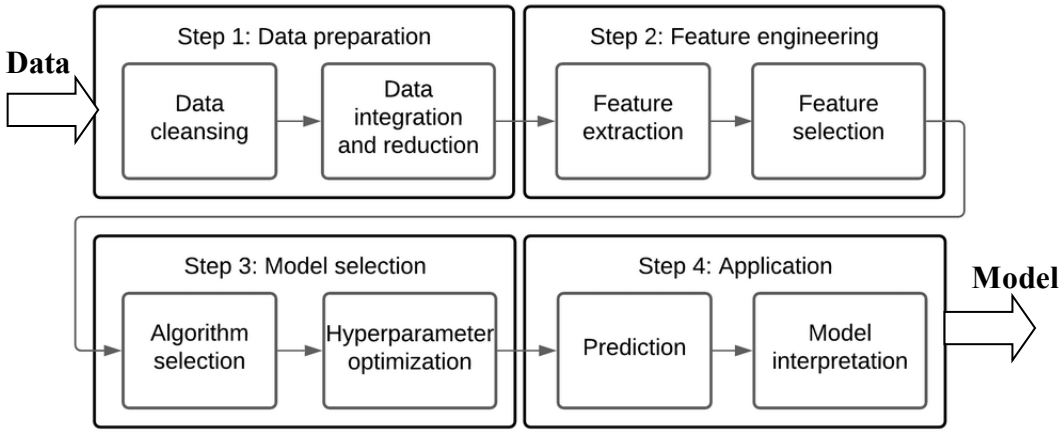

Рис.7. Загальна структура підготовки даних для побудови моделі ML.

Підготовка даних (Step 1). Це перший крок, де данні (Data) можуть збиратися з різних джерел, зокрема, в даній роботі – з платформи *Wireshark.* На основі збереженого трафіку у форматі *.CSV* формується DataFrame. Якщо такий DataFrame не містить ARP атак, тоді до нього додаються нові записи, які характерні для відповідних ARP атак. На цьому етапі відбувається очищення даних (Data Cleaning), видаляються або коригуються неправильні, відсутні або непотрібні дані (Missing value), відбувається фільтрація ARP-пакетів і залишаються тільки ті, які належать до протоколу ARP.

Конструювання ознак (Step 2). На цьому етапі відбувається вилучення зайвих ознак (*Feature Extraction*) і виділення необхідних (*Feature Selection*), в тому числі можуть створюватися нові. Якщо датасет містить багато ознак, які не є важливими для аналізу або моделювання, вони можуть бути видалені, щоб спростити аналіз, зменшити розмірність та обчислювальні витрати. Даний етап реалізується різноманітними методами аналізу даних, в тому числі кореляційними підходами.

Вибір і підготовка моделі (Step 3). На цьому етапі відбувається кодування категоріальних ознак (*Encoding Categorical Features*) для переведення їх в числовий формат, масштабування ознак (*Feature Scaling*) для забезпечення однакового діапазону значень і покращення швидкості навчання моделей. Окрім того, на цьому етапі відбувається вибір алгоритму (класифікатора) для побудови моделі та визначення оптимальних гіперпараметрів.

Останній етап (Step 4) призначений для формування моделі ML, збереження цієї моделі або декількох в залежності від потреб аналізу атак. Отримані моделі

використовуються для прогнозування різноманітних атак, на які вони були налаштовані та навчані.

На рис.8 представлений промаркований DataFrame на основі реальних даних, отриманих з платформи *Wireshark.* Як видно з даних, записи #56, #168 представляють *ARP spoofing* і вони, відповідно, промарковані як аномальні, де Out = «1». Якщо під час аналізу трафіку не було аномальних записів, то їх можна синтетично додати, створивши таким чином власний промаркований DataFrame з бажаним видами атак, які потрібно розпізнати.

No.	Time	Source	Destination	Protocol	Length	Info	Out
324	9.021666	72:7e:18:e7:d2:ef	Broadcast	ARP	60	Gratuitous ARP for 192.168.119.80 (Request)	0
54	4.646389	HewlettP_bf:91:ee	Dell_c0:56:f0	ARP	60	Who has 172.16.0.107? Tell 172.16.0.1	0
55	4.646442	Dell_c0:56:f0	HewlettP_bf:91:ee	ARP	42	172.16.0.107 is at 00:21:70:c0:56:f0	0
56	4.646455	HewlettP_bf:91:ee	Dell_c0:56:f0	ARP	60	172.16.0.1 is at 00:25:b3:bf:91:ee	1
166	6.744431	HewlettP_bf:91:ee	Dell_c0:56:f0	ARP	50	Who has 172.18.10.10? Tell 172.16.0.5	0
167	6.748552	Dell_c0:56:f0	HewlettP_bf:91:ee	ARP	88	172.18.10.10 is at 01:2f:20:ce:44:fa	0
168	6.750123	HewlettP_bf:91:ee	Dell_c0:56:f0	ARP	50	172.16.0.5 is at 34:56:af:31:9f:26	1
165	14.392559	HewlettP_bf:91:ee	Broadcast	ARP	60	Who has 172.16.0.1? Tell 172.16.0.105	0
166	4.646389	HewlettP_bf:91:ee	Dell_c0:56:f0	ARP	60	Who has 172.16.1.108? Tell 172.16.0.2	0

Рис.8. Сформований DataFrame на базі трафіку платформи *Wireshark.*

Отриманий DataFrame необхідно трансформувати в такий вид, щоб він був придатний для отримання моделі ML. В роботі пропонується модель на основі ключових ознак «Source», «Destination», «Info» та «Out», де «Out» - марковані мітки, які відповідають нормальному («0») і аномальному («1») трафіку.

В роботі розглядається задача побудови простої і ефективної моделі класифікації трафіка, тому застосовувалися класичні класифікатори ML, вбудовані до бібіліотеки *Sklearn Python*. Окрім того, при отриманні моделі класифікації потрібно враховувати часові залежності надходження пакетів, щоб не застосовувати складні нейромережеві структури типу LSTM *(Long Short-Term Memory*) та ін. Для цього запропоновано ряд функцій для побудови моделі формування нового DataFrame, придатного для застосування в ML:

- *ip_to_int(ip)* - приймає IP-адресу як рядок і повертає її числовий еквівалент. Якщо IP-адреса порожня або дорівнює '0.0.0.0', функція повертає 0;
- *mac_to_int(mac)* приймає MAC-адресу як рядок і повертає її числовий еквівалент. Якщо MAC-адреса порожня або дорівнює '00:00:00:00:00:00', функція повертає 0. Функція також замінює шістнадцяткові літери на відповідні числові значення перед обчисленням;
- використовуються регулярні вирази для аналізу IP- та MAC-адрес, додаються нові ознаки, такі як «who_has», «ip1», «tell», «ip2», «is_at», «ip_at», «mac», які містять відповідні частини ARP-пакетів;
- текстові частини, такі як «who_has», «tell», «is_at», кодуються у числовий формат за допомогою *LabelEncoder* для подальшого використання в алгоритмах ML;
- формується новий DataFrame з результатами для зберігання закодованих і числових значень;
- обробка пакетів відбувається циклічно по всіх рядках початкового DataFrame, аналізуючи відповідність «who_has», «tell», «is_at» ознаки «Info».

Результатом такої підготовки даних є новий DataFrame, представлений на рис.9. В ньому міститься перетворена інформація ключових ознак (рис.8), які можуть бути використані для побудови моделі ML (Step 3).

	who_has_encoded	ip1_int	tell_encoded	ip2_int	is_at_encoded	ip_at_int	mac_int	Label
0	0	0	0	0	0	0	0	0
1	1	2886729835	1	2886729729	1	2886729835	91376597142	0
2	0	0	0	0	1	2886729729	109343105158	1
3	1	2886863370	1	2886729733	1	2886863370	2023344714214	0
4	0	0	0	0	1	2886729733	37640944653082	1
...	...	...	...	...	...	...	...	...
210	1	3232235683	1	3232235550	0	0	0	0
211	1	3232235684	1	3232235550	0	0	0	0
212	1	2886731216	1	2886729985	1	2886731216	108573441259	0
213	0	0	0	0	1	2886729985	1115297164974881	1

Рис.9. Сформований DataFrame за результатом виконання конструювання ознак.

Для отримання моделей ML вхідні дані мережевого тарфіка (*Network Traffic*) утворюють Dataset, який підлягає попередній обробці (*Preprocessing*) згідно загальної структури на рис.7. Такий набір данних був розділений на тренувальний (*Train Data* - 70%) і тестовий (*Test Data* - 30%) набори (рис.10). Тренувальний набір використовується для отримання моделі класифікації для різних класифікаторів. В роботі було застосовано декілька класичних і добре відомих класифікаторів ML бібліотеки Sklearn, зокрема *DecisionTreeClassifier, LogisticRegression, SVM.* Навчані моделі були збережені для подальшого виклику і розгортання у форматі *.PKL*. Таким чином, для аналізу трафіку в реальному часі в подальшому відпадає необхідність поточного навчання моделі і витрата ресурсів і часу, що може бути досить вагомим аргументом для ресурсообмежених систем та мереж. Замість цього можна скористатися заздалегідь підготовленим набором моделей або їх комбінаціями в залежності від поставленої задачі.

Тестовий набір даних (*Test Data*) використовується для валідації якості отриманих моделей по ряду показників ефективності. Таким показниками слугують відомі характеристики: якість (*Accuracy*), влучність (*Precision*), повнота (*Recall*), *f1*-оцінка. Всі показники ефективності на тестовому наборі для вказаних класифікаторів показали максимальний результат – *1.0*, тобто сто відсоткову відповідність при менших часових ресурсах на навчання моделі у порівнянні із застосуванням нейромережевих моделей.

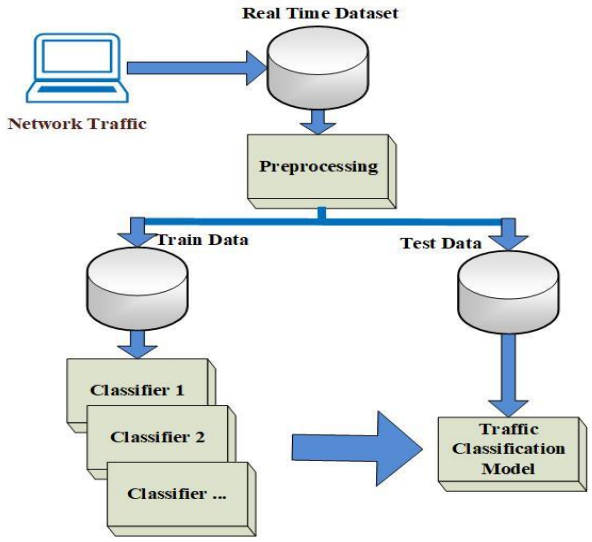

Рис.10. Модель побудови класифікаторів і валідації їх ефективності

Подальше вдосконалення моделей залежить від гіперпараметризації - налаштування гіперпараметрів моделі для досягнення найкращих результатів. При

потребі дані моделі можуть вдосконалюватися і підналаштовуватися під інші види мережевих атак без зміни апаратної частини і застосування коштовних контролерів SDN.

Серед останніх етапів виділяються наступні:

- впровадження моделі - перенесення навченої моделі в виробниче середовище для використання в реальних умовах згідно трафіку, отриманого з платформи *Wireshark;*
- моніторинг та підтримка - слідкування за продуктивністю моделі в реальному часі та оновлення її при необхідності.

4. Результати дослідження та обговорення. Впровадження отриманої моделі аналізу мережевого трафіка для аналізу ARP атак потребує розробки *системи автоматизованої підтримки* перетворення файлів .PCAPNG, які формує платформа *Wireshark* в режимі реального часу у формат .CSV. *Wireshark* має налаштування щодо формування файлів .PCAPNG заданих розмірів для їх подальшого збереження. Разом з тим *Wireshark* не передбачає можливості перетворення «на льоту» файлів у формат .CSV, що вимагає розробки додаткового методу автоматизованого перетворення. Отримані «на льоту» .CSV файли потребують попередньої підготовки, щоб до них можна було застосувати попередньо отриману модель ML для класифікації трафіка.

Система *автоматизованої підтримки ML моделі* складається з застосування декількох нових методів.

1. Метод перетворення .PCAPNG файлів платформи *Wireshark* у формат .CSV в режимі реального часу, який полягає в наступному:

- підготовка середовища - визначаються шляхи до папок з вхідними файлами .PCAPNG та вихідними файлами .CSV;
- визначається файл .CSV, який буде виключено з обробки (для усунення некоректної роботи у разі відсутності у трафіку ARP пакетів);
- аналіз пакетів – функція читає вхідний .PCAPNG файл та аналізує пакети для виявлення різних типів протоколів (Ethernet, ARP, IP, DHCP);
- збирання інформації про кожен пакет (джерело, призначення, тип протоколу, довжина та інші деталі);
- запис зібраної інформацію у вихідний .CSV файл.

Основні операції даного методу полягають у:

- отриманні списку всіх .PCAPNG файлів у папці з вхідними файлами (автоматично завантажуються з платформи *Wireshark*);
- визначення, які файли ще не були оброблені;
- циклічно перевіряє папку на наявність нових файлів, які автоматично записуються;
- для кожного нового файлу .PCAPNG викликається функція обробки та збереження результатів у новий .CSV файл;
- додає оброблені файли до списку оброблених, щоб уникнути повторної обробки;
- видаляє файл після обробки, якщо це не виключений файл (щоб не було перевантаження памяті).

2. Метод обробки DataFrame .CSV файла і формування заданих фітчів (ознак) для застосування підготовленої моделі ML, який полягає в наступному:

- фільтрація та вибір даних ARP - обирається підмножина даних, яка містить лише пакети з протоколом ARP;
- обробка текстових даних з ознакою «Info» – текст виділяється на окремі частини за допомогою регулярних виразів для отримання IP-адрес, MAC-адрес та інших даних;
- застосування функцій перетворення даних у числовий формат: IP-адресу та MAC-адресу;
- створення нового DataFrame для зберігання результатів нових перетворень;
- кодування текстових ознак у числові;

- заповнення та структурування результатів перетворення, які за своєю структурою повинні збігатися зі структурою, яка була використана для отримання відповідної моделі ML (рис.9);

Отже, цей метод дозволяє виконати обробку і трансформацію даних .CSV файла для подальшого машинного навчання і застосування підготовленої моделі ML.

3. Аналіз трафіка в режимі онлайн із застосуванням підготовленої моделі ML, який полягає в наступному:

- зчитуються дані з .CSV файлу (формуються на попереньому етапі) і до них додаються інші дані з підготовленого файлу addata.csv на той випадок, якщо у поточному трафіку відсутні APR запити для нормальної роботи моделі ML;
- завантажується підготовлена модель ML для прогнозування класу аномальності і застосовується для кожного окремого підготовленого .CSV файлу;
- виводяться кількісні характеристики аномального трафіку у кожному поточному файлі на екран з подачею звукового повідомлення чи передачею інформації на електронну пошту (при потребі).

Результат обробки файлів у випадку ARP атаки виводиться на екран (електронну пошту, сервер і т.п.) із позначенням номера пакета, який підпадає під атаку (рис.11).

```
File: 1   Number of abnormal traffic - 104
File: 2   Number of abnormal traffic - 203

 Completion of processing files in the directory
```

Рис.11. Результат повідомлення про аномальний трафік

Використання нових моделей машинного навчання для аналізу мережевого трафіку виявилося результативним у практичному застосуванні. Моделі дозволяють ефективно виявляти аномальний трафік у великих обсягах даних в реальному часі, що є критичним аспектом для забезпечення кібербезпеки в сучасних мережах. Основні переваги включають високу точність прогнозування аномальних подій, гнучкість у роботі з різноманітними форматами даних та ефективність у використанні ресурсів обчислювальних систем.

У порівнянні з нейромережевими моделями, які відомі своєю здатністю до автоматичного визначення складних зв'язків у даних, нова модель має певні переваги, які полягають у менших обчислювальних ресурсах для тренування та прогнозування, що робить її більш доступною для впровадження в системах з обмеженими ресурсами, наприклад IoT мережах без застосування складних контролерів SDN. Окрім того, отримані моделі зазвичай володіють більшою інтерпретованістю результатів, що дозволяє оперативно реагувати на виявлені аномалії та виправляти їх.

Впровадження автоматизованої системи аналізу трафіку мережі на базі запропонованих методів обробки даних та нових моделей машинного навчання є перспективним кроком у забезпеченні кібербезпеки та ефективного управління мережевими ресурсами, яке полягає у спрощенні реалізації, збільшенні ефективності та зниженні витрат, що робить її привабливим вибором для сучасних інформаційних технологій.

Висновки. Широке розповсюдження SDN та IoT мереж забезпечило високу гнучкість і ефективність керування ними, але водночас поставило нові виклики у захисті мережевої інфраструктури. Однією з важливих загроз залишаються атаки підробки протоколу розпізнавання адрес - ARP, що порушують цілісність мережі та конфіденційність даних. У цій роботі представлено новий підхід до виявлення ARP-спуфінгу в мережах, який враховує обмеження існуючих методологій і використовує методи машинного навчання ML для аналізу мережевого трафіку в реальному часі. Запропонований метод використовує дані, отримані з платформи Wireshark, і на основі машинного навчання класифікує та виявляє зловмисний мережевий трафік, що виникає

в результаті ARP-атак. Модель демонструє виняткову надійність, досягаючи 100% точності виявлення ARP-спуфінгу, що є критично важливим для підтримки швидкості реагування мережі. Використання методів ML дозволяє значно підвищити ефективність і швидкість виявлення загроз, забезпечуючи високий рівень безпеки мережевої інфраструктури. У порівнянні з нейромережевими моделями, метод машинного навчання має кілька переваг. Він вимагає менше обчислювальних ресурсів для тренування та прогнозування, що робить його доступнішим для впровадження в умовах обмежених ресурсів. Крім того, нова модель є більш інтерпретованою, що дозволяє оперативно реагувати на виявлені аномалії. Простота і швидкість впровадження, незалежність від інфраструктури, гнучкість та адаптивність до нових атак роблять цей підхід привабливим для широкого спектра організацій. Побудова моделі ML для виявлення ARP-атак на локальному комп'ютері також має свої переваги порівняно із застосуванням спеціалізованих контролерів SDN. Вона є ефективною в розподілених середовищах, де моделі можуть бути навчені локально на кожному комп'ютері або вузлі мережі, що дозволяє розподілене виявлення загроз без необхідності централізованого керування. Такий підхід забезпечує необхідний рівень безпеки мережі без значних інвестицій і складнощів, пов'язаних з SDN, що робить його доступним рішенням для багатьох організацій.

Список літератури

1. Odom W.: CCNA 200-301 Official Cert Guide. Volume 1-2. Cisco Press, 2019. 1095 p.
2. Carthern C., Wilson W., Bedwell R., Rivera N. Cisco Networks: Engineers' Handbook of Routing, Switching, and Security with IOS, NX-OS, and ASA. Apress Media, 2015. 839 p.
3. Santos O, Stuppi J. CCNA Security 210-260 Official Cert Guide. Apress Media, 2016. 608 p.
4. Микитишин А. Г., Митник М. М., Стухляк П. Д., Пасічник В. В.. Комп'ютерні мережі. Львів: Магнолія 2006, 2013. 256 с.
5. Sanders C. Practical packet analysis. Using Wireshark to solve real-world network problems. 2019. 448 p.
6. Bock L. A definitive guide to expertly analyzing protocols and troubleshooting networks using Wireshark. 2022. 606 p.
7. Girdler T., Vassilakis V.G.: Implementing an intrusion detection and prevention system using software-defined networking. Defending against ARP spoofing attacks and blacklisted MAC addresses *Comput. Electr. Eng.*, V. 90. 2021. DOI: 10.1016/j.compeleceng.2021.106990
8. AbdelSalam A.M., El-Sisi A.B.V., Reddy K. Mitigating ARP spoofing attacks in software-defined networks. *25th International Conference on Computer Theory and Applications, ICCTA*. 2015. P.. 126-131. DOI: 10.1109/ICCTA37466.2015.9513433
9. Amin R., Hussain M., Alhameed M., Raza S.M., Jeribi F., Tahir A.: Edge-computing with graph computation: A novel mechanism to handle network intrusion and address spoofing in SDN *Comput. Mater. Continua*. V.65 (3). 2020. P. 1869-1890. DOI: 10.32604/cmc.2020.011758
10. Aldabbas H., Amin R.: A novel mechanism to handle address spoofing attacks in SDN based IoT. *Cluster Comput.* V. 24 (4). 2021. P. 3011-3026. DOI: 10.1007/s10586-021-03309-0
11. Xu Y., Sun H., Xiang F., Sun Z.: Efficient ddos detection based on K-FKNN in software defined networks. *IEEE Access.* V. 7. 2019. P. 160536-160545. DOI: 10.1109/ACCESS.2019.2950945
12. Abdulla H., Al-Raweshidy H., Awad W. ARP Spoofing Detection for IoT Networks Using Neural Networks. *Proceedings of the Industrial Revolution & Business Management: 11th Annual PwR Doctoral Symposium (PWRDS*). 2020.
13. Shilpa P. Khedkar, Ramalingam A.C. Classification and Analysis of Malicious Traffic with Multi-layer Perceptron Model. *Ingénierie des Systèmes d'Information.* V. 26. No. 3. P. 303-310. 2021.
14. Hnamte V., Hussain J. Enhancing security in Software-Defined Networks: An approach to efficient ARP spoofing attacks detection and mitigation. *Telematics and Informatics Reports*, V. 14. 2024.

DEVELOPMENT OF AN ARTIFICIAL INTELLIGENCE METHOD FOR ANALYSIS OF HARMFUL NETWORK TRAFFIC AT CHANNEL LEVEL (ARP ATTACKS)

V.V. Palahin[1], O.A. Palahina[2], O.V. Ivchenko[3], O.M. Panasko[4], R.L. Ptashkin[5]

[1-4]Cherkasy State Technological University
460, Shevchenko blvd., Cherkasy, Ukraine, 18005
[5]Cherkasy scientific research forensic centre MIA of Ukraine
104, Pasterivska st., Cherkasy, Ukraine, 18000
emails: palahin@ukr.net[1], palahina@ukr.net[2],
sania_ivchenko@ukr.net[3] , lena.pa@ukr.net[4], ndekc.ck@gmail.com[5]

The widespread distribution of software-defined networks (Software-Defined Networking - SDN) and IoT networks has provided flexibility and efficiency in network management. However, it has also posed new challenges in protecting network infrastructure. Address Resolution Protocol (ARP) spoofing attacks, which violate network integrity and data confidentiality, remain one of the significant threats. This manuscript presents a new approach to detecting ARP spoofing in networks, addressing the limitations of existing methodologies. The analysis of ARP protocols, their purposes, and basic methods of protection against attacks was carried out. Typical threats to computer networks at the physical and data link layers of the OSI model are presented, along with an analysis of the features of detecting such threats using artificial intelligence (AI) methods. The application of machine learning (ML) methods for traffic analysis based on real-time data from the Wireshark platform is proposed. The new method uses AI to classify and detect malicious network traffic generated by ARP protocol attacks. The developed model and method demonstrate exceptional robustness, achieving 100% ARP spoofing detection accuracy, which is critical for maintaining network responsiveness. The analysis results can be used to make informed decisions about the choice of protection methods for networks with different purposes and information protection requirements. Using AI to monitor and analyze network traffic can significantly increase the effectiveness and speed of threat detection. Due to its ability to adapt to new types of attacks and detect more complex anomaly patterns, the proposed approach provides a higher level of network infrastructure security. This research demonstrates the potential of innovative technologies in the fight against cyber threats and contributes to the development of reliable protection methods for modern networks.

Keywords: network traffic analysis, ARP Spoofing, Artificial Intelligence, L2 Level Attacks