ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ КІБЕРБЕЗПЕКИ ТА ЗАХИСТУ ІНФОРМАЦІЇ

КАФЕДРА СИСТЕМ ТА ТЕХНОЛОГІЙ КІБЕРБЕЗПЕКИ

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Засоби протидії фішинговим атакам електронної пошти корпоративної системи на базі рішення SpamAssassin»

зі спеціальності *125 Кібербезпека*
(код, найменування спеціальності)

освітньо-професійної програми *Інформаційна та кібернетична безпека*
(назва програми)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ Володимир РЕШЕТНИК
(підпис)

Виконав: здобувач вищої освіти групи БСД-41
РЕШЕТНИК Володимир
(прізвище, ім'я)

Керівник к.т.н., доц. БОРСУКОВСЬКИЙ Юрій
(науковий ступінь, вчене звання, прізвище, ім'я)

Рецензент _____
(науковий ступінь, вчене звання, прізвище, ім'я)

Київ 2025

ВСТУП

Актуальність дослідження. Фішинг є однією з найбільш поширених та руйнівних кіберзагроз, що використовує соціальну інженерію для маніпуляції користувачами та викрадення конфіденційної інформації або встановлення шкідливого програмного забезпечення. Його ефективність полягає у використанні людських вразливостей, а не лише технічних недоліків систем. Корпоративний сектор є особливо вразливим, оскільки фішингові атаки можуть призвести до значних фінансових втрат, витоків даних, репутаційних збитків та операційних перебоїв.

Вразливість корпоративного середовища до фішингу є не тільки технічною, а й глибоко психологічною та організаційною. Успішність фішингових атак, попри наявність технічних засобів захисту, свідчить про те, що людський фактор є фундаментальним викликом, який не може бути повністю усунений лише технологічними рішеннями. Це вимагає комплексного підходу, що поєднує технічні засоби, навчання персоналу та вдосконалення внутрішніх процесів.

Apache SpamAssassin є провідною відкритою платформою для боротьби зі спамом, яка використовується системними адміністраторами для класифікації та блокування небажаних електронних листів. Вона застосовує широкий спектр тестів, включаючи аналіз заголовків та вмісту, байєсівську фільтрацію, DNS-чорні списки та бази даних колаборативної фільтрації. SpamAssassin відмінно виявляє фішингові повідомлення, які намагаються обманом виманити фінансову інформацію. Його модульна архітектура дозволяє легко інтегрувати інші технології для боротьби зі спамом та фішингом.

Об'єкт дослідження – захист корпоративної електронної пошти від фішингових атак.

Предмет дослідження – методи та засоби протидії фішинговим атакам на основі рішення SpamAssassin.

Мета роботи – розробити варіант системи фільтрації електронної пошти на

базі рішення SpamAssassin та надати рекомендації щодо застосування методів захисту від фішингових атак.

Наукові завдання:

Провести теоретичний аналіз природи фішингових атак як одного з основних векторів соціальної інженерії в корпоративному середовищі.

Узагальнити існуючі підходи, методи та засоби виявлення та блокування фішингових повідомлень у системах електронної пошти.

Дослідити архітектуру, принципи функціонування та механізми виявлення фішингового контенту, реалізовані у системі фільтрації SpamAssassin.

Оцінити ефективність виявлення фішингових загроз на основі правил, сигнатур і методів аналізу вмісту, які використовуються у SpamAssassin.

Розробити структурну модель інтеграції рішення SpamAssassin у корпоративну інфраструктуру електронної пошти з урахуванням вимог інформаційної безпеки.

Сформулювати практичні рекомендації щодо налаштування, супроводу та оптимізації системи захисту від фішингових атак на основі SpamAssassin для корпоративного використання.

Методи дослідження – аналіз наукових публікацій, технічної документації, порівняння функціоналу захисних систем, вивчення стандартів інформаційної безпеки, логічне моделювання.

Практичне значення одержаних результатів:

Розроблено рекомендації щодо впровадження рішення SpamAssassin у корпоративну систему електронної пошти з метою протидії фішинговим атакам. Надано практичні поради для фахівців з кібербезпеки щодо налаштування, моніторингу та оновлення системи фільтрації з урахуванням специфіки корпоративного середовища.

Результати кваліфікаційної роботи апробовані на Всеукраїнській науково-практичній конференції «Цифрова трансформація кібербезпеки», яка відбулася 24 квітня2025 року в Державному університеті інформаційно-комунікаційних технологій, м.Київ.

1 АНАЛІЗ ПРОБЛЕМИ ВИЯВЛЕННЯ ФІШИНГОВИХ АТАК В КОРПОРАТИВНІЙ ЕЛЕКТРОННІЙ ПОШТІ

1.1. Проблеми виявлення фішингу у корпоративному середовищі

Проблематика виявлення фішингових атак у корпоративному середовищі є складною та охоплює як технічні, так і поведінково-психологічні й організаційні аспекти. Однією з основних вразливостей залишається людський фактор, експлуатований через методи соціальної інженерії. Фішинг, як форма соціальної інженерії, орієнтується не на технічні вади, а на особливості когнітивної поведінки людини. Зловмисники навмисно впливають на емоційну сферу жертви, активуючи такі тригери, як страх, терміновість, зацікавленість або очікування вигоди, що істотно знижує здатність до критичного аналізу. Навіть добре підготовлені користувачі не завжди в змозі адекватно оцінити ризики, оскільки фішингові атаки постійно модифікуються, а наявність знань не гарантує їх своєчасне застосування. Організаційні політики, зокрема ті, що регламентують зміну паролів або порядок використання IT-систем, часто не компенсують вплив імпульсивних рішень, що виникають унаслідок неуважності або сприйняття повідомлення як авторитетного.

Ефективність фішингових атак пояснюється не лише технологічними прогалинами, а й наявністю стійких когнітивних упереджень та організаційних недоліків, які формують основу експлуатаційної вразливості. Зловмисники активно використовують такі упередження, як упередження підтвердження, евристику доступності та схильність до підкорення авторитетам. Ці когнітивні механізми значно підвищують ймовірність ірраціональних дій з боку жертви у стресових або ресурсно обмежених умовах. Відсутність критичного осмислення в умовах тиску часу створює поведінковий розрив між обізнаністю користувача щодо загроз та фактичними діями у момент прийняття рішень. Такий когнітивно-поведінковий дисонанс є стійкою вразливістю, що ускладнює реалізацію заходів інформаційної безпеки на людському рівні.

Еволюція фішингових методів у бік персоналізації та технологічної складності унеможливлює ефективне виявлення цих атак винятково за допомогою традиційних засобів фільтрації. Техніки масової розсилки трансформувалися у високотаргетовані сценарії, включно з атаками на топменеджмент (whaling), цільовими атаками (spear phishing) та компрометацією ділової пошти (BEC). Такі атаки ґрунтуються на попередньому дослідженні цільових осіб та використанні персоналізованих повідомлень із точним відтворенням мовного стилю або комунікаційної манери. Зловмисники розширюють вектори впливу через канали голосового фішингу (vishing), SMS-комунікації (smishing), QR-коди (quishing) та застосування сервісів класу Phishing-as-a-Service (PhaaS), що істотно знижує бар'єр входу для атаки навіть з боку недостатньо кваліфікованих осіб.

Складність фішингових атак зростає не лише кількісно, а й якісно — особливо в контексті впровадження генеративного штучного інтелекту. Сучасні AI/LLM-моделі дають змогу створювати повідомлення з високим ступенем персоналізації, відтворювати стиль листування конкретних осіб або навіть імітувати голоси чи відеозображення. Використання таких технологій усуває звичні індикатори шахрайства — граматичні помилки, стилістичні аномалії або неприродну побудову фраз. Додаткову складність становить використання "довірених" каналів комунікації (зокрема, корпоративних хмарних сервісів, DNSSEC), які маскують злочинну активність під легітимну. Унаслідок цього традиційні сигнатурні підходи дедалі частіше демонструють низьку ефективність, і захист електронної пошти вимагає переходу до моделей, заснованих на поведінковому аналізі, контекстуальній валідації та адаптивному реагуванні.

Наслідки реалізованих фішингових атак для корпоративного середовища є системними та багатофакторними. Фінансові втрати можуть включати як прямі збитки у вигляді несанкціонованих переказів або виплат викупу, так і непрямі витрати, пов'язані з відновленням систем, юридичним супроводом, аудитом інциденту та підвищенням страхових внесків. Компрометація конфіденційних даних призводить до втрати комерційної інформації, розголошення внутрішніх комунікацій і, як наслідок, до порушення нормативних вимог (GDPR, HIPAA, PCI

DSS) та накладення санкцій. Додаткові наслідки включають дестабілізацію бізнес-процесів через необхідність ізоляції уражених систем, тимчасове блокування доступу для користувачів та проведення масштабного реагування. Порушення репутаційної цілісності організації тягне за собою втрату довіри з боку партнерів, зменшення клієнтської бази та скорочення доходів у довгостроковій перспективі.

Окрему загрозу для ефективного функціонування систем захисту становить феномен "втоми від сповіщень" (alert fatigue), що проявляється в десенсибілізації персоналу до великої кількості тривожних сигналів. Надмірна генерація сповіщень, значна частка яких є хибнопозитивними або неінформативними, суттєво ускладнює розпізнавання дійсно критичних інцидентів. Це призводить до зниження швидкості реагування, пропуску загроз та розбалансування роботи SOC-команд. Крім того, зловмисники можуть навмисно генерувати потоки подій, імітуючи широкомасштабну активність, щоб маскувати справжню атаку серед фонових сповіщень. Така ситуація створює системну вразливість, коли обсяг попереджень суперечить можливості їх оперативної обробки [1].

1.2. Основні техніки фішингових атак та проблеми їх ідентифікації

Фішингові атаки демонструють постійну еволюцію, адаптуючись до нових технологічних умов і вдосконалюючи методи ухилення від виявлення та впливу на кінцевих користувачів. Сучасний фішинг уже давно перестав бути лише технічною проблемою — він охоплює широкий спектр соціотехнічних підходів, що поєднують маніпуляцію свідомістю, обхід засобів захисту й адаптацію до поведінки жертви.

Штучний інтелект докорінно змінив фішинг, дозволяючи генерувати високоваріативні та невловимі кампанії в великих масштабах. У 2024 році значні 90,9% поліморфних атак, проаналізованих KnowBe4, демонстрували певне використання штучного інтелекту, а 76,4% усіх фішингових атак містили принаймні одну поліморфну ознаку. Ці кампанії характеризуються серією майже ідентичних електронних листів, які відрізняються лише незначними, ледь

помітними деталями, що є свідомим рішенням, спрямованим на обхід традиційних механізмів виявлення "відомих шкідливих" зразків.

Зловмисники ретельно змінюють різні компоненти електронної пошти, включаючи назви шкідливих вкладень, відображувані імена відправників, підписи електронної пошти, адреси електронної пошти або метадані зображень. Серед найпоширеніших змін – заміна логотипів організацій, зміна цільових посилань або зміна домену електронної адреси відправника. Зловмисники стратегічно додають зайві символи та знаки або змінюють довжину та шаблони тексту в темах листів; ця тактика розроблена для уникнення блокування рідними засобами безпеки та захищеними поштовими шлюзами (SEGs). Крім того, розміщення випадкових символів у кінці тем листів може ефективно приховувати їх за межами обрізки попереднього перегляду електронної пошти. Значна частина (36,9%) поліморфних атак використовує невидимі символи для порушення роботи підходів виявлення на основі штучного інтелекту, таких як обробка природної мови (NLP) та розуміння природної мови (NLU). Ці символи, хоча й непомітні для людського ока, читаються програмним забезпеченням для виявлення та можуть використовуватися для розділення слів або маніпулювання гіперпосиланнями, створюючи сліпі зони для виявлення. Більшість (52%) поліморфних фішингових електронних листів надходять з компрометованих облікових записів, за якими слідують фішингові домени (25%) та веб-пошта (20%), що є стратегією, яка використовується для обходу перевірок автентифікації домену. Штучний інтелект має можливість запобігати ідентичності двох електронних листів, програмно створюючи відмінний вміст електронної пошти для кожного одержувача. Ця динамічна генерація надзвичайно ускладнює для засобів безпеки, таких як захищені поштові шлюзи (SEGs), ідентифікацію послідовних шаблонів або сигнатур, які зазвичай застосовуються для виявлення фішингових атак. Зловмисники використовують складні методи, включаючи шаблонні двигуни зі змінною заміною, генерацію тексту за допомогою ланцюгів Маркова для природних варіацій, алгоритми генерації доменів (DGAs) для систематичного створення нових доменів, а також

бібліотеки обфускації для динамічної зміни коду та URL-адрес через кілька рівнів кодування або структурної модифікації.

Фішингові атаки все частіше слугують основним та високоефективним вектором для доставки корисних навантажень програм-вимагачів, причому з вересня 2024 року по лютий 2025 року спостерігалося значне збільшення на 22,6% кількості програм-вимагачів, доставлених фішинговими електронними листами. Ця тривожна тенденція прискорюється, про що свідчить збільшення на 85,6% використання HTML-контрабанди, поширеної техніки обфускації, лише між листопадом 2024 року та лютим 2025 року [3].

Технічні механізми: HTML-контрабанда є високоефективною технікою доставки шкідливого програмного забезпечення, яка використовує легітимні функції HTML5 та JavaScript для "контрабанди" закодованих шкідливих скриптів у спеціально створених HTML-вкладеннях або веб-сторінках. Шкідливі дані не передаються безпосередньо через мережу у своїй сирій, виявлюваній формі; натомість, вони конструюються на стороні клієнта, у веб-браузері користувача. Ця збірка на стороні клієнта ефективно обходить мережеві брандмауери та традиційні рішення безпеки, включаючи пісочниці та антивірусні двигуни в застарілих проксі-серверах, які в основному перевіряють дані під час передачі. Зловмисники вбудовують бінарні файли як закодований текст у вихідний код HTML, який потім браузер реконструює у виконуваний файл шкідливого програмного забезпечення на операційній системі хоста. Поширені тактики включають використання об'єктів HTML5 Blob для інкапсуляції даних та кодування Base64 для перетворення бінарних даних у текстовий формат, який може оброблятися JavaScript. Складні корисні навантаження програм-вимагачів, такі як INC Ransom, проаналізовані у наданому звіті, використовують кілька рівнів обфускації. Ці рівні часто включають захищені паролем zip-файли у поєднанні з HTML-контрабандою, що є комбінацією, розробленою для уникнення механізмів виявлення на основі сигнатур, присутніх у рідних рішеннях безпеки Microsoft 365 та захищених поштових шлюзах (SEGs). Випадково згенерований текст стратегічно додається до корисних навантажень для штучного збільшення часу сканування та для

заплутування як автоматизованих сканерів безпеки, так і людських аналітиків. Навіть незначні зміни цього тексту-заповнювача можуть значно змінити криптографічний хеш скрипта, тим самим роблячи виявлення на основі сигнатур неефективним для подальших ітерацій того ж шкідливого програмного забезпечення. Кіберзлочинці маніпулюють URL-рядками за допомогою передових методів, таких як зворотний скрипт (відображення тексту задом наперед) та кодування Base64. Потім вони фрагментують обфускований URL на непослідовні сегменти, розкидані в HTML-скрипті, що є методом, розробленим для приховування справжнього URL від чорних списків. Після активації скрипт динамічно повертає ці методи для реконструкції оригінального шкідливого URL. Зловмисники стратегічно збільшують розміри файлів для вкладень шкідливого програмного забезпечення та шкідливих HTML. Ця тактика використовується для активації угод про рівень обслуговування (SLA) затримки електронної пошти, що дозволяє шкідливому навантаженню обійти початкові механізми виявлення, які працюють у межах певних часових обмежень, перш ніж поштова система повністю обробить та просканує вміст.

Хоча електронна пошта беззаперечно залишається домінуючим вектором для фішингових атак, складаючи приблизно 96% усіх фішингових атак та слугуючи точкою входу для приблизно 91% усіх кібератак , зловмисники все частіше диверсифікують свої канали зв'язку. Ця стратегічна диверсифікація має на меті уникнути встановлених механізмів виявлення та використовувати нові, менш захищені вразливості на різних платформах.

Технічні механізми та вектори доставки: Смішинг (SMS-фішинг) становив приблизно 28% фішингових атак станом на 2023 рік, причому майже 40% усіх загроз для мобільних пристроїв конкретно стосувалися фішингу облікових даних через SMS. Зловмисники надсилають шахрайські текстові повідомлення, часто видаючи себе за легітимні джерела, такі як банки, служби доставки або урядові установи, які направляють одержувачів на шкідливі веб-сайти або спонукають їх надавати конфіденційну інформацію безпосередньо через текст. Атаки смішингу зазнали значного зростання на 22% лише в третьому кварталі 2024 року.

Користувачі зазвичай виявляють вищий ступінь довіри до текстових повідомлень порівняно з електронними листами та мають менше доступних інструментів для перевірки легітимності на мобільних пристроях, що сприяє ефективності смішингу. Вішинг (голосовий фішинг) переживає значний сплеск, причому кількість зареєстрованих інцидентів зросла більш ніж на 16% у четвертому кварталі 2023 року порівняно з попереднім кварталом, і на приголомшливі 260% порівняно з четвертим кварталом 2022 року. Зловмисники видають себе за персонал технічної підтримки, представників служби підтримки клієнтів або інші легітимні організації, щоб витягти конфіденційну інформацію по телефону. Передовий вішинг, керований штучним інтелектом, може включати високоінтерактивні сценарії дзвінків, де система динамічно реагує на запити жертв, що робить їх майже невідрізними від справжніх розмов. Робоча група з боротьби з фішингом (APWG) випустила попередження щодо наближення "епохи вішингу, керованого штучним інтелектом", що характеризується високою реалістичністю можливостей клонування голосу. Квішинг (QR-код фішинг) є новою технікою, яка використовує зростаючу залежність суспільства від QR-кодів для різних повсякденних дій. Шахраї вбудовують шкідливі QR-коди у фішингові електронні листи (часто як зображення або вбудовані в PDF-файли) або розповсюджують їх фізичними засобами (наприклад, фальшиві наклейки на паркоматах). При скануванні ці коди перенаправляють користувачів на шахрайські веб-сайти, ретельно розроблені для крадіжки облікових даних, фінансових даних або для сприяння встановленню шкідливого програмного забезпечення. Традиційні сканери безпеки часто мають труднощі з ефективним читанням QR-кодів порівняно зі звичайними текстовими посиланнями, що робить їх ефективним методом обходу фільтрів URL-адрес. QR-коди, згенеровані штучним інтелектом, можуть бути створені для направлення цілей на фішингові веб-сайти без негайного виникнення підозр. Гібридні "зворотні" шахрайства (TOAD) є складними атаками, які стратегічно поєднують електронну пошту та телефонну тактику. Вони зазвичай починаються з електронного листа, що містить лише номер телефону, спонукаючи користувача зателефонувати до, здавалося б, легітимної, але насправді фальшивої служби підтримки. Цей

багатоканальний метод розроблений для того, щоб спонукати жертв розголошувати конфіденційну інформацію або надавати віддалений доступ по телефону, причому в середньому 10 мільйонів таких спроб відбувається щомісяця. Фішингова діяльність також все частіше спостерігається в ширшому спектрі платформ цифрового зв'язку, включаючи програми для відеоконференцій (що впливає на 44% організацій), програми для ділових чатів (40%), служби обміну файлами (40%) та різні інші платформи обміну повідомленнями. Прогнози свідчать, що майбутні атаки можуть додатково використовувати голосових помічників або пристрої Інтернету речей (IoT) як нові канали для фішингу. Швидкий прогрес технології глибоких фейків (deepfake video technology) також створює значну проблему, дозволяючи створювати фальшивих учасників в онлайн-зустрічах для оманливих цілей.

Зловмисники все частіше застосовують високоцільові тактики соціальної інженерії, часто доповнені штучним інтелектом, щоб ретельно видавати себе за довірених осіб або організації та таким чином обманом змушувати жертв компрометувати конфіденційну інформацію або вчиняти шкідливі дії.

Технічні механізми та вектори доставки: На відміну від широкомасштабних фішингових кампаній, цільовий фішинг (spear phishing) ретельно розроблений для націлювання на конкретних осіб або організації. Зловмисники проводять широку розвідку, збираючи детальну інформацію (наприклад, через профілі в соціальних мережах, публічні записи, корпоративні веб-сайти) для створення високоперсоналізованих та винятково переконливих повідомлень. Ці електронні листи можуть тонко посилатися на нещодавні проекти, спільні зв'язки або конкретні посади, щоб значно підвищити їхню достовірність та релевантність для цілі. Незважаючи на те, що вони становлять мізерну частку (менше 0,1%) від загального обсягу електронної пошти, електронні листи цільового фішингу непропорційно ефективні, складаючи значні 66% усіх витоків даних. Китобійний фішинг (whaling) є спеціалізованою та дуже потужною формою цільового фішингу, яка націлена виключно на високопоставлених осіб, таких як топ-менеджери або ключові особи, що приймають рішення в організації. Зловмисники приділяють

значний час та зусилля збору розвідувальних даних та складній соціальній інженерії, прагнучи створити винятково переконливі повідомлення, які використовують авторитет та доступ цілі. Ці атаки помітно зросли, використовуючи залежність від віддалених робочих середовищ. Компрометація ділової електронної пошти (BEC) передбачає складне видавання себе за колегу або керівника (часто топ-менеджера або довіреного постачальника), щоб обманом змусити співробітників здійснювати несанкціоновані банківські перекази, розголошувати конфіденційну інформацію або змінювати законні платіжні реквізити. Ці атаки майстерно використовують вроджену довіру та сприйнятий авторитет, пов'язані з певними електронними адресами, що робить їх особливо небезпечними. Шахрайства BEC часто передбачають ретельне планування, причому зловмисники вивчають ієрархії компанії та схеми комунікації для створення дуже правдоподібних та переконливих запитів. Фінансовий вплив шахрайств BEC є приголомшливим, спричинивши збитки в розмірі приблизно 50,8 мільярда доларів США між 2013 та 2022 роками. Критичною характеристикою електронних листів BEC є їхня часта відсутність шкідливих посилань або вкладень, що дозволяє їм обходити традиційні сканування на наявність шкідливого програмного забезпечення. Штучний інтелект все частіше використовується для покращення якості перекладу та для створення більш переконливих та контекстуально відповідних запитів, особливо в неангломовних регіонах. Зомбі-фішинг є підступною технікою, яка передбачає використання скомпрометованих облікових записів, що продовжують функціонувати без відома законного користувача. Алгоритми штучного інтелекту аналізують минулі комунікації, щоб визначити оптимальні моменти для вставки шкідливих повідомлень та виявити найефективнішу мову для отримання бажаної відповіді, таким чином безперешкодно інтегруючись у поточні, законні потоки комунікації. Технологія глибоких фейків: Генеративні моделі штучного інтелекту навчаються імітувати точну мову та стилі написання як окремих осіб, так і корпоративних організацій, охоплюючи все: від офіційного тону корпоративної службової записки до невимушеного стилю нотатки колеги. Ця можливість поширюється на генерацію

високореалістичних голосових записів і навіть відео глибоких фейків, що надзвичайно ускладнює для людей розрізнення їх від справжніх комунікацій. Яскравим прикладом реального впливу технології відео глибоких фейків була її ймовірна роль у соціальній інженерній атаці на Arup вартістю 20 мільйонів фунтів стерлінгів. Кіберзлочинці стратегічно впроваджуються в процес найму, використовуючи сфабриковані резюме, фотографії, маніпульовані штучним інтелектом, та викрадені номери соціального страхування. Вони використовують законні процеси подання заяв на роботу, вимагаючи від зацікавлених сторін завантажувати документи, що містять шкідливі посилання або програмне забезпечення, часто націлюючись на інженерні посади через пов'язану з ними мобільність робочих місць та привілейований доступ до систем. Шкідливі завдання з кодування, замасковані під законні завдання, та використання спільних поштових скриньок також є поширеними тактиками [3].

Ідентифікація фішингових атак ускладнюється через широкий спектр технічних методів ухилення від виявлення. Зокрема, активно застосовується обфускація URL-адрес із використанням гомогліфів, символів з різних алфавітів (наприклад, кириличних символів у латинському контексті), доменних скорочувачів (типу bit.ly), IP-адрес замість доменів, символу «@» для редиректу, HTML-сутностей та нетипових портів. Такі механізми суттєво ускладнюють розпізнавання навіть для обізнаного користувача. Додатково використовується обфускація HTML та CSS-коду, яка включає приховані стилі, нульову ширину символів, прозорі елементи або багатомовне наповнення, що обходить текстові фільтри систем безпеки.

Іншим вектором ухилення є стеганографія — приховування шкідливого контенту у зображеннях. Такий підхід дозволяє обійти сигнатурні фільтри та системи оптичного розпізнавання, особливо коли застосовується додаткова візуальна рандомізація або CAPTCHA-подібне спотворення. Поліморфне шкідливе програмне забезпечення постійно змінює свою структуру, уникаючи виявлення антивірусами, які базуються на статичних сигнатурах. Атаки нульового дня, які використовують ще не задокументовані вразливості, становлять окремий клас

небезпек, оскільки системи захисту не мають для них оновлених механізмів реагування.

Платформи PhaaS пропонують повністю готові інструменти для фішингових кампаній, включаючи шаблони повідомлень, техніки обходу захисту та навіть засоби обходу багатофакторної автентифікації. Це значно спрощує реалізацію атак для менш досвідчених зловмисників та підвищує загальний рівень загрози.

Додатковим ускладненням є використання довірених інфраструктур, таких як хмарні сервіси (OneDrive, SharePoint), для розміщення шкідливого контенту. Ці платформи часто мають чинні сертифікати безпеки, зокрема SSL, що створює помилкову впевненість у легітимності ресурсу. Це трансформує класичні «червоні прапорці» в «зелені», вимагаючи від систем захисту глибшого аналізу контексту та поведінки.

Психологічна складова фішингових атак ґрунтується на когнітивних упередженнях і емоційних реакціях. Зловмисники маніпулюють такими ефектами, як підтвердження власної думки, доступність уявлення про події та підкорення авторитетам. Сценарії, засновані на страху, терміновості або обіцянці вигоди, перекривають здатність жертви до критичного аналізу. Тимчасові обмеження, навмисне створені атакувальником, слугують каталізатором для прийняття імпульсивних, необґрунтованих рішень, що і забезпечує ефективність атаки.

Таким чином, сучасні фішингові кампанії характеризуються високим рівнем гнучкості, контекстної адаптації, технічної складності та психологічного впливу, що потребує від систем захисту багаторівневого підходу з орієнтацією на поведінкову аналітику, машинне навчання та глибоку перевірку контенту поза межами класичних сигнатурних механізмів [2].

1.3. Аналіз існуючих рішень для виявлення фішингових атак електронної пошти корпоративної системи

Сучасні антифішингові рішення стикаються зі значними технічними обмеженнями, які перешкоджають їхній здатності ефективно протистояти

динамічній еволюції фішингових загроз. Традиційні методи виявлення, такі як сигнатурні підходи та чорні списки, хоча й є основоположними, виявляються недостатніми проти нових та постійно змінюваних тактик атак. Ці статичні методи, що покладаються на заздалегідь визначені шаблони або відомі шкідливі об'єкти, не можуть адаптуватися до швидких змін, що призводить до високого рівня хибних спрацьовувань (false positives) та пропущених атак (false negatives).

Обмеження, пов'язані зі штучним інтелектом та машинним навчанням: Хоча системи виявлення фішингу на основі машинного навчання (ML) та глибокого навчання (DL) демонструють значні перспективи у класифікації загроз на основі вмісту, URL-адрес та репутації відправника, вони мають власні недоліки. Одним із головних обмежень є проблема "чорного ящика", де ці моделі не надають чіткого обґрунтування своїх рішень. Ця непрозорість підриває довіру користувачів та аналітиків безпеки, що зменшує ймовірність того, що вони довірятимуть попередженням або вживатимуть відповідних заходів. Крім того, ці моделі вимагають великих обсягів позначених навчальних даних, які не завжди легкодоступні, та є обчислювально інтенсивними, що обмежує їхню масштабованість та розгортання в реальному часі. Проблема дисбалансу даних, коли легітимних зразків значно більше, ніж фішингових, також може призвести до упередженості моделей та зниження ефективності виявлення.

Обхід багатофакторної автентифікації (MFA): Навіть впровадження MFA, яке часто вважається надійним засобом захисту, не є панацеєю. Зловмисники розробили складні методи обходу MFA, зокрема атаки "людина посередині" (Adversary-in-the-Middle, AiTM), які використовують зворотні проксі-сервери для перехоплення облікових даних та токенів сесії. Ці атаки дозволяють зловмисникам обійти MFA, оскільки вони отримують дійсні токени сесії, які дозволяють їм отримувати доступ до облікових записів без необхідності повторного проходження MFA. Техніки, такі як ін'єкція JavaScript та перевірка IP-адреси/User-Agent, використовуються для уникнення виявлення, що робить ці атаки надзвичайно складними для виявлення традиційними засобами безпеки.

Обхід шлюзів безпеки електронної пошти: Фішингові електронні листи, згенеровані штучним інтелектом, можуть бути ретельно спроектовані для обходу фільтрів, які блокують спам та шкідливі вкладення, імітуючи законні протоколи зв'язку та уникаючи стандартних протоколів автентифікації, таких як DMARC. Ця здатність до адаптації означає, що технічні засоби захисту повинні постійно еволюціонувати, щоб не відставати від нових тактик зловмисників.

Ці виклики підкреслюють концепцію "динамічного ворожого навчання". Зловмисники постійно адаптують свої тактики, щоб уникнути виявлення, створюючи безперервну гонку озброєнь у кібербезпеці. Традиційні системи виявлення, що базуються на статичних правилах або відомих сигнатурах, не можуть ефективно протистояти цій адаптивній загрозі. Це призводить до високих показників хибних спрацьовувань та хибних негативів, що підриває довіру до систем безпеки та створює операційні проблеми. Сучасні фішингові атаки, особливо ті, що використовують штучний інтелект, не просто еволюціонують; вони активно навчаються на невдалих спробах, щоб покращити свої методи обходу. Наприклад, поліморфний фішинг, керований штучним інтелектом, динамічно змінює елементи електронної пошти, щоб уникнути виявлення на основі шаблонів. Це означає, що захисні системи повинні бути не просто реактивними, а проактивно адаптивними, здатними до безперервного навчання та оновлення своїх моделей виявлення в реальному часі. Вимога до "пояснюваного штучного інтелекту" (XAI) стає критичною, оскільки, щоб користувачі та аналітики довіряли та ефективно використовували складні системи виявлення на основі штучного інтелекту, вони повинні розуміти логіку, що стоїть за рішеннями системи. Без цієї прозорості впровадження передових технічних рішень буде обмеженим. Таким чином, майбутнє виявлення фішингу вимагає розробки адаптивних, пояснюваних систем на основі штучного інтелекту, які можуть не лише виявляти нові загрози, але й надавати чіткі, дієві пояснення, щоб подолати розрив між автоматизованим виявленням та довірою користувачів.

Протидія фішинговим атакам в електронній пошті корпоративного сегмента вимагає комплексного, багатошарового підходу, оскільки жодне з існуючих рішень

не забезпечує абсолютного захисту від усього спектру загроз. Сучасні технології виявлення фішингу охоплюють широкий спектр інструментів — від фільтрації трафіку та перевірки автентичності повідомлень до поведінкового аналізу й засобів реагування, — що дозволяє формувати стійку модель інформаційної безпеки організації.

Перший рівень захисту зазвичай реалізується через шлюзи безпеки електронної пошти (Secure Email Gateways), які аналізують вхідну та вихідну кореспонденцію на предмет шкідливого вмісту, спам-компонентів та підозрілої поведінки. Ці системи здатні проводити глибоку перевірку вкладень і гіперпосилань, використовують алгоритми машинного навчання для розпізнавання фішингових шаблонів, а також реалізують блокування або карантин підозрілих повідомлень. Незважаючи на високу ефективність на етапі доставки, шлюзи не завжди здатні виявляти загрози, що проявляються після того, як лист потрапив на кінцеву точку користувача, особливо за умови використання обфускації, складних технік обходу та експлуатації нульових вразливостей.

Антивірусні рішення, хоч і залишаються важливим елементом системи захисту, демонструють обмежену ефективність у виявленні нових варіантів шкідливого програмного забезпечення, які змінюють свій код або структуру задля уникнення сигнатурного виявлення. Це особливо стосується поліморфного шкідливого ПЗ, що активно застосовується у фішингових кампаніях для доставки шкідливих завантажень або активації шкідливих скриптів через легітимні на вигляд канали комунікації.

Інструменти класу EDR (Endpoint Detection and Response) забезпечують постійну видимість подій на рівні кінцевих точок, дозволяючи аналітикам виявляти ознаки компрометації, слідкувати за поведінкою процесів та виконувати зворотній аналіз інцидентів. Вони збирають системні логи, події мережевого трафіку та дії користувачів, що дозволяє виявляти аномалії, які не виявляються через звичайну фільтрацію. Проте, фокусування EDR винятково на кінцевих точках обмежує огляд подій у хмарних застосунках, API або системах електронної пошти, де також реалізуються фішингові вектори атак.

Платформи класу XDR (Extended Detection and Response) усувають ці обмеження, забезпечуючи кореляцію телеметрії з різнорідних джерел: мереж, кінцевих точок, хмарних середовищ, систем автентифікації та електронної пошти. Завдяки єдиній моделі даних XDR дозволяє не лише виявляти складні багатокрокові атаки, але й автоматизувати реагування на них, що критично важливо для виявлення фішингу, який діє у поєднанні з соціальною інженерією, імперсонацією та BEC.

У межах традиційних систем фільтрації, таких як Apache SpamAssassin, реалізовано адаптивну модульну архітектуру з відкритим кодом, яка дозволяє оперативно впроваджувати нові правила виявлення, реагуючи на зміну тактик зловмисників. Проте максимальна ефективність SpamAssassin досягається при його інтеграції з рішеннями постдоставкової обробки, зокрема з EDR або XDR-платформами. Це дозволяє покрити весь життєвий цикл фішингової атаки — від перехоплення листа до реагування на наслідки на кінцевій точці або в хмарі.

Іншою важливою категорією є технології автентифікації електронної пошти. Протоколи SPF, DKIM та DMARC відіграють базову роль у захисті від спуфінгу, надаючи змогу перевірити справжність домену відправника. Проте, вони мають суттєві обмеження. Зокрема, SPF не підтримує сценарії з переадресацією, DKIM обмежений підписом лише частини повідомлення, а DMARC не враховує візуально схожі домени, що дозволяє зловмисникам обходити захист через використання "майже ідентичних" імен. Також ці протоколи не захищають у разі компрометації легітимного облікового запису. Це зумовлює необхідність доповнення автентифікації контентним та поведінковим аналізом.

Проактивне виявлення фішингу потребує інтеграції з платформами аналізу загроз (Threat Intelligence Platforms), які забезпечують актуальну інформацію про IP-адреси, домени, сигнатури та поведінкові патерни, що притаманні фішинговим кампаніям. Інтеграція індикаторів загроз у рішення SIEM, EDR або XDR дозволяє підвищити точність виявлення та зменшити кількість хибних спрацювань, а також автоматизувати процес реагування.

Не менш важливою складовою стратегії захисту є навчання користувачів. Регулярні симуляції фішингових атак, навчальні курси та інтерактивні модулі підвищують рівень обізнаності персоналу, формуючи навички розпізнавання ознак фішингу. Це є особливо актуальним, оскільки ефективність фішингових кампаній багато в чому базується на експлуатації людської неуважності, авторитету та емоційного стану жертви.

Додатково до цього, системи запобігання витоку даних (DLP) забезпечують моніторинг переміщення чутливої інформації та блокування несанкціонованих передач даних, які можуть бути ініційовані після успішного фішингу. Такі рішення реалізують контроль доступу до документів, їх шифрування та політики доступу, знижуючи ризик витоків конфіденційної інформації.

Впровадження засобів багатофакторної автентифікації (MFA) у поєднанні з системами управління ідентичністю (IAM) значно знижує ймовірність несанкціонованого доступу навіть у разі компрометації облікових даних. Найбільш ефективними є MFA-рішення, що використовують криптографічні методи без паролів, прив'язку до домену та перевірку контексту автентифікації.

Таким чином, ефективна протидія фішинговим атакам в електронній пошті потребує цілісного підходу, що поєднує шлюзові фільтри, поведінкову аналітику, автентифікаційні протоколи, проактивний моніторинг, навчання користувачів та контроль над інформаційними потоками. Впровадження багаторівневої архітектури захисту забезпечує стійкість до як масових, так і цільових фішингових атак, що постійно удосконалюються технічно та психологічно [5].

Для комплексного аналізу ефективності засобів виявлення фішингових атак в електронній пошті доцільно порівняти можливості провідних комерційних продуктів з відкритим рішенням Apache SpamAssassin. Такий підхід дозволяє не лише визначити функціональні відмінності, а й оцінити переваги з точки зору гнучкості, інтеграції, економічної доцільності та адаптивності до нових загроз.

Microsoft 365 Defender, інтегрований у хмарну платформу Microsoft 365, забезпечує багатоаспектний захист за рахунок поєднання алгоритмів машинного навчання, сигнатурного аналізу та політик автентифікації. Серед його ключових

функцій — виявлення імперсонації, перевірка справжності відправників, глибоке сканування вкладень і посилань, а також визначення рівнів чутливості за допомогою гнучкої системи порогових значень. Крім того, реалізовано модуль виявлення компрометації ділової пошти (BEC), що є критично важливим для запобігання фінансовим шахрайствам. Однак, ефективність системи залежить від регулярного оновлення механізмів фільтрації, а також від швидкості адаптації до нових технік ухилення, зокрема обфускації, шкідливого вмісту в мультимедійних об’єктах або AI-генерованих повідомлень. Незважаючи на тісну інтеграцію з іншими сервісами Microsoft, система залишається вразливою до методів, що обходять традиційні сигнатури або експлуатують довіру до внутрішніх повідомлень.

Google Workspace забезпечує захист електронної пошти, базуючись на розширених інструментах аналізу поведінки, штучного інтелекту та глибокого сканування вмісту. Важливою функцією є Enhanced Safe Browsing, що дозволяє виявляти потенційно небезпечні посилання до моменту їх відкриття. Розширене сканування вкладень із застосуванням Security Sandbox забезпечує додатковий рівень перевірки файлів у контрольованому середовищі. Водночас інтеграція з Chrome Enterprise дозволяє поширити фільтрацію на зовнішні вебресурси, а впровадження багатофакторної автентифікації підвищує рівень захисту облікових записів. Обмеженням залишається залежність від адміністративних прав для конфігурації повного набору функцій, а також часткова недоступність механізмів перевірки при активному шифруванні даних на стороні клієнта, що обмежує можливості Safe Browsing у Gmail.

SpamTitan є спеціалізованим рішенням для захисту електронної пошти, яке поєднує методи аналізу вмісту, перевірку автентичності відправників та евристичний аналіз. Його функціональність включає виявлення спроб імперсонації, аналіз гіперпосилань, перевірку вкладень у пісочниці та реалізацію політик SPF, DKIM, DMARC. Технології машинного навчання й грейлістинг підвищують точність виявлення, тоді як гнучке налаштування політик дозволяє адаптувати систему під потреби конкретної організації. Однак це рішення є

комерційним продуктом, що потребує фінансових витрат, а також зберігає певну залежність від вендора в аспекті оновлень і підтримки.

Система N-able Mail Assure забезпечує хмарний захист електронної пошти з орієнтацією на колективний аналіз загроз. Інструмент підтримує захист у режимі реального часу, довготривале архівування, інтеграцію з Microsoft 365, а також навчання системи через зворотній зв'язок від користувачів. Завдяки використанню пропрієтарної технології Intelligent Protection and Filtering, рішення демонструє високу точність виявлення. Проте його комерційний характер і закритість алгоритмів можуть бути обмеженням для організацій, які віддають перевагу прозорим системам безпеки.

На тлі розглянутих комерційних рішень Apache SpamAssassin вирізняється відкритістю, гнучкістю, високою кастомізованістю та економічною ефективністю. Як система з відкритим вихідним кодом, SpamAssassin не потребує ліцензійних платежів, що робить його привабливим вибором для організацій з обмеженим бюджетом або для тих, хто прагне контролювати всі аспекти безпеки самостійно. Модульна архітектура дозволяє створювати, редагувати та розгортати індивідуальні правила для виявлення нових або специфічних атак, а також інтегрувати сторонні плагіни для розширення функціональності.

Окремою перевагою є прозорість роботи: адміністратор має повний доступ до логіки класифікації повідомлень, вагових коефіцієнтів і критеріїв оцінки. Це забезпечує не лише високу керованість, а й здатність проводити глибоку діагностику у випадках помилкової класифікації. Крім того, SpamAssassin підтримує автоматичне оновлення правил (через sa-update) та може інтегруватися з відомими джерелами індикаторів загроз (OpenPhish, PhishTank), що забезпечує своєчасне оновлення механізмів виявлення. Завдяки активній спільноті проєкт постійно розвивається та залишається релевантним щодо сучасних векторів атак, включаючи AI-генерований фішинг і шкідливі вкладення зі складною структурою.

Певні обмеження пов'язані з відсутністю графічного інтерфейсу та необхідністю кваліфікованого обслуговування. Однак вони можуть бути зменшені шляхом інтеграції з панелями управління сервером (cPanel, Plesk) або

впровадженням клієнт-серверної архітектури spamd/spamc для оптимізації масштабованості та продуктивності. Враховуючи це, Apache SpamAssassin може виступати як самостійне рішення або як ключовий компонент багаторівневої архітектури виявлення фішингу, особливо в середовищах, де цінуються контроль, гнучкість, прозорість і можливість швидкої адаптації до змін [4].

2 АНАЛІЗ МЕТОДІВ ТА ЗАСОБІВ ВИЯВЛЕННЯ ТА ФІЛЬТРАЦІЇ ФІШИНГУ НА ПРИКЛАДІ SPAMASSASSIN

2.1. Архітектура та функціональні можливості системи SpamAssassin

SpamAssassin є інтелектуальним фільтром електронної пошти, орієнтованим на виявлення небажаних повідомлень шляхом багаторівневої перевірки їхнього змісту та заголовків. Основу його функціонування становить застосування великої кількості тестів, кожен з яких відповідає за виявлення специфічних ознак спаму або фішингових повідомлень. Результати таких перевірок обробляються через статистичну модель, що дозволяє формувати комплексну оцінку ризику повідомлення.

Архітектура системи побудована за модульним принципом із використанням гнучкого фреймворку оцінювання, системи плагінів та засобів розширення. Такий підхід забезпечує високий рівень адаптивності — нові функціональні компоненти можуть бути додані до системи без порушення її базової логіки. Збереження конфігурацій у текстовому форматі сприяє простоті розширення функціоналу через правила й плагіни, а також прискорює реакцію на появу нових форм фішингових атак.

Ключовим компонентом архітектури є фреймворк оцінювання, у якому кожне правило має ваговий коефіцієнт — позитивний або негативний. У процесі аналізу повідомлення до нього застосовуються всі доступні правила, після чого підраховується сумарна оцінка. У разі перевищення певного порогового значення повідомлення класифікується як спам. Така система дозволяє тонко налаштовувати чутливість фільтра для різних сценаріїв, забезпечуючи адаптацію до потреб конкретного інформаційного середовища.

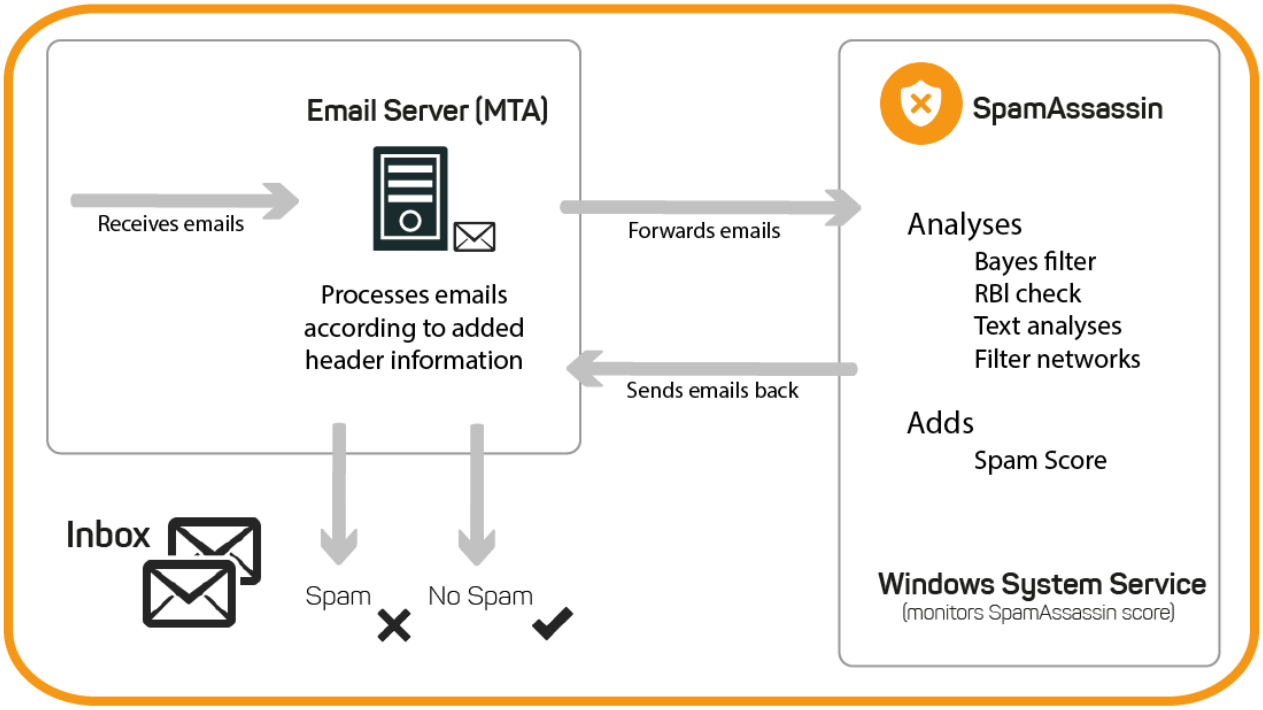


Рис.2.1. Архітектура SpamAssassin

Модульна побудова SpamAssassin сприяє динамічному оновленню та швидкому реагуванню на нові загрози. Здатність інтегрувати плагіни, додавати або модифікувати правила в режимі реального часу, а також регулярні оновлення, роблять цю систему не статичним фільтром, а адаптивним механізмом протидії сучасним загрозам. У контексті розповсюдження поліморфного та AI-генерованого фішингу така адаптивність є принципово важливою для підтримання високої ефективності.

Процес обробки повідомлень у SpamAssassin відбувається послідовно та охоплює декілька етапів. Після надходження листа на поштовий сервер запускається механізм аналізу, який спершу перевіряє адресу відправника на відповідність довіреним доменам, репутацію джерела та інші релевантні характеристики. Далі здійснюється поглиблений аналіз змісту листа: виявлення тригерних фраз, типових для спаму, аналіз форматування, перевірка наявності прихованих гіперпосилань, оцінка структури HTML та перевірка вмісту на предмет нетипових мовних конструкцій. Усі ці характеристики враховуються при формуванні остаточної оцінки повідомлення, яка визначає його класифікацію.

Якщо сума балів перевищує заданий поріг (як правило, 5.0), лист маркується як спам. Подальші дії визначаються політиками організації — повідомлення може бути переміщене до спеціальної папки або позначене відповідним заголовком.

У великих інфраструктурах критичною є продуктивність фільтрації, і для її оптимізації SpamAssassin використовує модель клієнт-сервер із застосуванням компонентів spamd і spamc. Демон spamd виконує роль постійного сервісу, який завантажує правила лише один раз та обслуговує всі вхідні запити. Це дозволяє уникнути повторної ініціалізації кожного разу при обробці нового повідомлення, зменшуючи навантаження на ресурси та пришвидшуючи час реакції. Клієнт spamc має мінімальну ресурсну вартість і призначений для передавання листів до демонічного процесу, що забезпечує мінімізацію затримок і високий рівень масштабованості.

Збереження ядра SpamAssassin у пам’яті підвищує ефективність використання системних ресурсів і виключає необхідність повторного завантаження всіх компонентів. У контексті розподілених середовищ це дозволяє забезпечити відмовостійкість та балансування навантаження — клієнт може автоматично перемикатися між доступними серверами spamd. Крім того, за допомогою параметра --headers клієнт має змогу обробляти лише заголовки листа, що знижує вимоги до пропускної здатності мережі. Це особливо важливо у великих організаціях, де обробляються десятки або сотні тисяч листів щодня.

Таким чином, SpamAssassin являє собою адаптивну, масштабовану та технічно досконалу систему виявлення небажаного електронного трафіку, яка поєднує гнучкість конфігурації, інтеграцію з іншими інструментами та високу ефективність обробки великих обсягів поштових повідомлень. Його архітектурна гнучкість та можливості динамічного розширення роблять систему придатною до використання в умовах еволюціонуючих загроз, зокрема у боротьбі з сучасними формами фішингових атак [6].

2.2. Алгоритми виявлення фішингових повідомлень у SpamAssassin

Система фільтрації SpamAssassin реалізує багаторівневий підхід до виявлення фішингових повідомлень, що включає комбінацію правилового аналізу, статистичних методів, колаборативних мереж, репутаційних сервісів та інтеграцію з динамічними джерелами загроз. Такий підхід забезпечує як виявлення масових, так і цільових фішингових кампаній, включаючи атаки, що використовують техніки обфускації або генеруються за допомогою засобів штучного інтелекту.

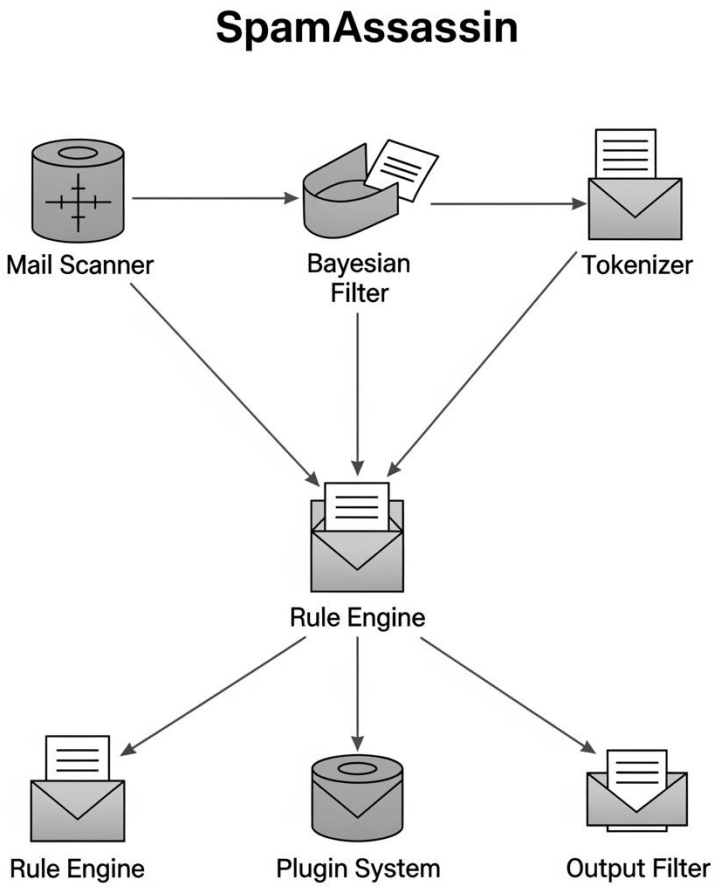


Рис.2.2. Ключові компоненти SpamAssassin

Ключовим компонентом системи є евристичний аналіз, що ґрунтується на великій кількості правил, реалізованих через регулярні вирази. Аналіз охоплює заголовки, тіло повідомлення, структуру HTML, використання ключових фраз, граматичні патерни, стилістичні аномалії та інші індикатори. Система надає

кожному правилу ваговий коефіцієнт, результати якого агрегуються для обчислення сумарного балу ризику. Повідомлення, що перевищують певний поріг, класифікуються як небажані. Разом з тим, ефективність статичних евристик знижується у випадках використання AI/LLM для генерації контенту без типових помилок чи шаблонів, що традиційно слугували індикаторами. Крім того, зловмисники активно застосовують HTML-обфускацію, шифрування URI, нестандартні кодування та структурні зміни повідомлення, що ускладнює застосування класичних правил. Це зумовлює необхідність постійного оновлення наборів правил, а також доповнення функціоналу поведінковими та контекстно-залежними методами виявлення.

Іншою фундаментальною складовою є байєсівська фільтрація. Її механізм базується на ймовірнісному аналізі вмісту повідомлень шляхом зіставлення з навчальними вибірками легітимної (ham) та шкідливої (spam) кореспонденції. Алгоритм виконує обчислення ймовірності на основі частоти появи певних слів або комбінацій, асоційованих зі спамом. Для надійної роботи фільтра потрібні збалансовані навчальні набори з не менше 200 повідомлень у кожній категорії. Навчання реалізується за допомогою інструменту sa-learn, який також дозволяє проводити перенавчання у випадках помилкової класифікації. Однією з ключових проблем залишається вразливість до отруєння навчальної бази (Bayesian poisoning), коли спамери свідомо додають до повідомлень нехарактерну лексику, що вводить фільтр в оману. Це вимагає впровадження механізмів контролю якості навчальних даних, регулярного аудиту фільтра та доповнення алгоритмів більш стійкими до маніпуляцій методами машинного навчання.

Важливу роль у виявленні фішингу відіграє перевірка репутації відправника за допомогою DNS-базованих чорних списків (DNSBLs) та перевірка URL-адрес, виявлених у повідомленні, через URIBL- та SURBL-служби. Ці сервіси надають в реальному часі інформацію про домени, що раніше були використані для розсилки шкідливої пошти або фішингових повідомлень. SpamAssassin автоматично аналізує гіперпосилання в тілі повідомлення та зіставляє їх з чорними списками. У разі збігу застосовується штрафний бал до загальної оцінки. Оскільки більшість фішингових

кампаній базується на переходах за URL, перевірка за допомогою URIBL/SURBL є однією з найефективніших превентивних методик. Водночас традиційні чорні списки мають затримку оновлення, тому для виявлення атак нульового дня SpamAssassin також інтегрується з динамічними фідами загроз.

Підсистема колаборативної фільтрації реалізується через інтеграцію з мережами Razor2, Pyzor та DCC. Razor2 — це розподілена система, яка використовує сигнатурний аналіз та колективний внесок користувачів для створення централізованої бази спаму. Pyzor працює аналогічно, обчислюючи дайджести повідомлень та співставляючи їх із хешами в базі. Служба DCC виконує агрегацію контрольних сум для виявлення масових розсилок. Використання цих механізмів дозволяє оперативно виявляти вже відомі шаблони шкідливих листів, незалежно від їх вмісту або мови, а також зменшувати навантаження на локальні ресурси завдяки централізованій перевірці. Незначна затримка в обробці компенсується підвищенням точності виявлення [7].

Окрему цінність становить підтримка динамічної інтеграції із зовнішніми фідами аналізу загроз, такими як OpenPhish та PhishTank. Завдяки плагіну Mail::SpamAssassin::Plugin::Phishing, система отримує дані про фішингові домени з інтервалами оновлення від 1 до 6 годин. Це дозволяє фільтру виявляти навіть ті загрози, що мають короткий життєвий цикл, властивий сучасним фішинговим кампаніям. Наприклад, середній час життя шкідливого домену складає лише кілька годин, тому використання статичних DNSBL уже не є достатнім. Динамічні фіди суттєво підвищують актуальність і релевантність рішень, дозволяючи оперативно реагувати на нові вектори атак.

SpamAssassin також підтримує розробку спеціалізованих правил, орієнтованих на виявлення ознак фішингу, включаючи URL-адреси, що використовують IP-замінники доменів, скорочувачі посилань або гомогліфи (наприклад, через Punycode). Аналіз HTML здійснюється з урахуванням прихованого вмісту, стилів, що маскують текст (через CSS-властивості display:none, visibility:hidden, font-size:0, color:transparent) та невидимих символів

(Zero Width Space). Такі методи обфускації дозволяють зловмисникам обходити традиційні фільтри, тому їх виявлення є критичним компонентом фільтрації.

Інструментарій дозволяє виявляти й фішингові шаблони, орієнтовані на конкретні бренди, наприклад, Apple, PayPal або Google. Виявлення таких шаблонів ґрунтується на наявності фраз, типових для відповідного інтерфейсу, візуальних стилів або URL-адрес, які імітують домени популярних сервісів. Додаткове правило TO_CC_IN_BODY, що перевіряє наявність адрес отримувача в тілі повідомлення, дозволяє виявити фішинг, що маскується під персоналізоване звернення. Для запобігання хибним спрацьовуванням правило додатково перевіряє, чи не міститься в тілі повідомлення адреса самого відправника, що часто трапляється у легітимних підписах.

Таким чином, ефективність SpamAssassin у виявленні фішингових повідомлень зумовлена комплексним поєднанням методів сигнатурного, евристичного, статистичного та колаборативного аналізу, а також можливістю розширення через інтеграцію з актуальними джерелами загроз. Ця архітектура дозволяє гнучко адаптувати систему до нових векторів атак і підтримувати високу ефективність фільтрації в умовах еволюції тактик соціальної інженерії.

2.3. Інтеграція SpamAssassin у корпоративну поштову інфраструктуру

SpamAssassin базується на гнучкій, модульній архітектурі, яка дозволяє інтегрувати різноманітні технології для ефективної боротьби зі спамом та фішингом. Основними виконавчими компонентами системи є spamassassin.exe, що виконує перевірку електронної пошти та присвоює їй спам-бал; spamd.exe, який є демонізованою версією spamassassin і працює як постійний фоновий процес, що слухає вхідні запити, забезпечуючи значно вищу ефективність при обробці великих обсягів пошти порівняно з одноразовим запуском spamassassin для кожного повідомлення; та spamc.exe, який виступає як клієнт для spamd, передаючи йому повідомлення для аналізу та повертаючи результат. Ефективність spamd полягає в

тому, що він завантажує правила лише один раз при запуску, що мінімізує накладні витрати на обробку кожного окремого листа.

Механізм оновлення правил реалізується за допомогою утиліти sa-update.exe, яка відповідає за завантаження найновіших наборів правил SpamAssassin. Цей процес є критично важливим і має виконуватися регулярно для забезпечення актуальності фільтра та його здатності протистояти новим загрозам. Правила SpamAssassin зберігаються у форматі звичайного тексту, що спрощує їх конфігурацію, модифікацію та додавання нових користувацьких правил. Інструмент sa-learn.exe призначений для навчання баєсівського фільтра SpamAssassin, використовуючи вибірки спам- та не-спам-повідомлень (так званих "ham" повідомлень). Це дозволяє системі адаптуватися до унікальних особливостей поштового трафіку конкретного середовища, значно підвищуючи точність класифікації та зменшуючи кількість хибних спрацьовувань. Баєсівський фільтр вивчає частоту появи певних "токенів" (слів або фраз) у спам- та не-спам-повідомленнях, а потім використовує ці дані для розрахунку ймовірності того, що нове повідомлення є спамом.

Архітектура SpamAssassin дозволяє інтегрувати різноманітні плагіни для розширення функціональності. Серед підтримуваних плагінів є AWL (для нормалізації оцінок через авто-білий список), AutoLearnThreshold (для автоматичного навчання баєсівського фільтра на основі порогів), Bayes (основний баєсівський класифікатор), DKIM (для перевірки автентифікації DKIM), SPF (для перевірки SPF-записів), URIDNSBL (для перевірки URL-адрес за DNS-чорними списками), Razor (для інтеграції з розподіленою мережею виявлення спаму) та DCC (Distributed Checksum Clearinghouse). Ці плагіни дозволяють SpamAssassin застосовувати широкий спектр тестів, включаючи аналіз заголовків, вмісту, URL-адрес, а також використовувати зовнішні бази даних та механізми спільної фільтрації. Кожен тест, що спрацьовує, додає або віднімає бали від загального спам-балу повідомлення, формуючи комплексне судження про його "спамність" [8].

Інтеграція SpamAssassin у поштову інфраструктуру є ключовим етапом для забезпечення ефективної фільтрації. Вибір методу інтеграції залежить від використовуваного агента передачі пошти (MTA) та специфічних вимог до продуктивності та гнучкості.

Для Postfix, одного з найпопулярніших MTA, інтеграція може бути реалізована через перенаправлення вхідної пошти через зовнішній скрипт або програму, яка передає її SpamAssassin для аналізу та модифікації. Цей скрипт, як правило, використовує spamc для взаємодії з spamd для підвищення продуктивності. Після перевірки, повідомлення може бути перенаправлене назад до Postfix для подальшої доставки або відхилене, якщо його спам-бал перевищує встановлений поріг.

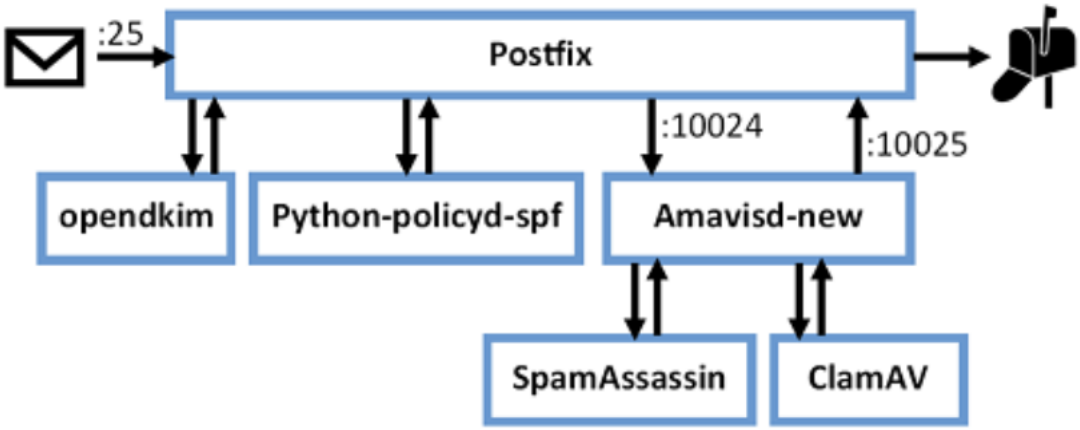


Рис.2.3. Ітеграція з Postfix

Інші методи інтеграції Postfix включають:

Amavisd-new: Це високоефективний інтерфейс на Perl, який розташовується між MTA та одним або кількома засобами перевірки вмісту, такими як SpamAssassin та антивіруси. Amavisd-new може працювати в режимах фільтрації до або після черги, причому фільтрація до черги є більш ефективною для відхилення спаму на ранніх етапах. Інтеграція Amavisd-new є більш складною, але дозволяє також підключати антивірусне сканування, наприклад, ClamAV.

SpamAssassin-milter: Це легкий компонент, що функціонує як milter-додаток, перехоплюючи повідомлення в SMTP-сесії MTA. Він дозволяє додавати заголовки

SpamAssassin (X-Spam-*), перезаписувати певні заголовки або навіть замінювати тіло повідомлення, якщо воно ідентифіковано як спам. SpamAssassin-milter часто є кращим за Amavis у деяких випадках, оскільки він за замовчуванням зменшує кількість зворотних розсилок (backscatter emails) майже до нуля, тоді як популярні інтеграції Amavis часто налаштовані в режимі фільтрації після черги, що може призвести до генерації таких розсилок.

spampd: Це фільтруючий SMTP-проксі, який використовується для фільтрації пошти між зовнішнім MTA/реле та "внутрішнім" MTA. Він пропонує більш надійне та ефективне рішення, ніж пряме використання spamd для Postfix, і підтримує фільтрацію як до, так і після черги.

Інтеграція зі Sendmail часто здійснюється шляхом конфігурації Sendmail для використання procmail як локального агента доставки, з подальшим додаванням рецепту procmail для маркування спаму у файл /etc/procmailrc. Цей підхід відрізняється простотою налаштування та дозволяє використовувати spamd і spamc для швидшої перевірки, а також задіювати файли користувацьких налаштувань та баєсівські бази даних.

Альтернативні методи інтеграції включають використання різних milter-інтерфейсів:

MIMEDefang: Sendmail milter, написаний на Perl, який фільтрує всі електронні листи, що надсилаються через SMTP, блокуючи віруси та спам за допомогою SpamAssassin. Він дозволяє відхиляти повідомлення з кодами помилок 5xx під час SMTP-сесії, що інформує відправника без генерації додаткового листа.

spamass-milt: Ще один Sendmail milter, який дозволяє фільтрувати спам безпосередньо в конвеєрі обробки повідомлень Sendmail.

milter-spamd та milter-spamc: Прості milter-інтерфейси для Sendmail.

Інтеграція з Exim

Для Exim існує кілька варіантів інтеграції:

Компіляція з WITH_CONTENT_SCAN=yes: Починаючи з версії Exim 4.50, можлива компіляція з цією опцією, що дозволяє Exim відхиляти спам після зчитування тіла, але до підтвердження прийняття електронної пошти.

Інтеграція як транспорту Exim: Цей метод передбачає додавання спеціальних маршрутизаторів та транспортів до конфігурації Exim для перенаправлення пошти через SpamAssassin. Це дозволяє видаляти заголовки X-Spam-* з вхідної пошти та додавати нові після перевірки.

Завантажувані модулі або заміни local_scan.c: Такі як SA-Exim, які дозволяють відхиляти спам до того, як він буде прийнятий MTA. На Debian це часто є методом за замовчуванням, що вимагає встановлення пакету sa-exim та редагування /etc/exim4/sa-exim.conf.

Інтеграція з Microsoft Exchange Server зазвичай вимагає використання сторонніх рішень, оскільки Exchange не має вбудованої прямої підтримки SpamAssassin. Прикладами таких рішень є Exchange Server Toolbox та Exchange SpamAssassin Sink.

Exchange Server Toolbox: Функціонує як плагін для Microsoft Exchange Server, інтегруючи SpamAssassin та ClamAV для захисту від спаму та вірусів, а також забезпечуючи архівацію електронної пошти. Він діє як транспортний агент, обробляючи вхідні та вихідні повідомлення за допомогою гнучкої системи правил до того, як вони досягнуть поштових скриньок.

Spamassassin Agent for Microsoft Exchange Server: Це транспортний агент, розроблений на C#, який взаємодіє безпосередньо з Exchange, передаючи повідомлення клієнту spamc для оцінки spamd. Якщо оцінка перевищує поріг відхилення, агент позначає повідомлення заголовком X-Spam-Discard: YES, дозволяючи Exchange виконувати подальші дії. Транспортні агенти в Exchange Server використовують SMTP-події, які спрацьовують під час руху повідомлень по конвеєру транспортування, надаючи агентам доступ до повідомлень у певні моменти SMTP-сесії та маршрутизації. Наприклад, подія OnEndOfData спрацьовує, коли віддалений SMTP-хост завершує передачу даних повідомлення [9].

3 РОЗРОБКА РЕКОМЕНДАЦІЙ ЩОДО ПРОТИДІЇ ФІШИНГУ В ЕЛЕКТРОННІЙ ПОШТІ НА БАЗІ SPAMASSASSIN

3.1. Варіант розгортання SpamAssassin у корпоративній середовищі

SpamAssassin є багатоспектральним інструментом для фільтрації спаму, який використовує комбінацію різноманітних технік для досягнення високої точності класифікації електронних листів. Цей підхід робить його стійким до постійно змінюючихся тактик спамерів. Система застосовує широкий спектр локальних та мережевих тестів для ідентифікації спам-сигнатур, що ускладнює для спамерів розробку методів обходу фільтра. Вона аналізує як заголовки, так і вміст електронних листів, використовуючи розширені статистичні методи для їх класифікації. До ключових методів, які використовує SpamAssassin, належать текстовий аналіз, що виявляє підозрілі слова, фрази та шаблони у тексті повідомлення , а також баєсівська фільтрація, яка використовує статистичний аналіз для визначення ймовірності того, що повідомлення є спамом, ґрунтуючись на його вмісті. Ця вбудована здатність до навчання є критично важливою, оскільки вона дозволяє системі покращувати свою точність з часом, зменшуючи залежність від статичних правил та адаптуючись до нових загроз. Крім того, SpamAssassin використовує DNS-чорні списки (DNSBL) для перевірки IP-адрес відправників та доменів на відповідність відомим спискам джерел спаму , а також бази даних спільної фільтрації, такі як Pyzor, DCC та Vipul's Razor, які збирають інформацію про спам від великої кількості користувачів. Сучасні підходи до фільтрації спаму все частіше використовують моделі машинного навчання для ідентифікації шаблонів у даних та адаптації до нових типів спаму, що значно покращує продуктивність фільтрів. Об'єднання різних методів, наприклад, SVM з Naive Bayes, помітно підвищує точність класифікації. Це підкреслює, що ефективність SpamAssassin походить від його здатності інтегрувати різноманітні методи,

створюючи багатошаровий захист, де жоден окремий метод фільтрації не є достатнім для боротьби зі складним спамом.

Кожен електронний лист, що проходить через SpamAssassin, отримує числовий бал, який відображає ймовірність того, що це спам. Цей бал є сумою балів, нарахованих або віднятих за спрацювання різних правил та тестів. Кожне правило в SpamAssassin має асоційоване числове значення: позитивні числа вказують на можливий спам, тоді як негативні числа свідчать про те, що лист навряд чи є спамом. За замовчуванням, поріг для класифікації листа як спаму встановлено на рівні 5.0 балів. Якщо сума балів листа перевищує цей поріг, він позначається як спам. Системні адміністратори мають можливість налаштовувати цей поріг, роблячи фільтр більш або менш чутливим. Вищий бал означає вищу ймовірність того, що лист є спамом. Для корпоративного середовища, де помилкові спрацювання (false positives) можуть мати значні наслідки (наприклад, втрата важливих ділових комунікацій), розуміння та точне налаштування цієї системи оцінювання є критично важливим.

Розгортання та ефективне управління SpamAssassin у корпоративному середовищі вимагає врахування низки специфічних вимог та розуміння унікальних викликів, які виходять за рамки простої технічної конфігурації. Корпоративні системи фільтрації спаму повинні відповідати високим стандартам, щоб забезпечити безперебійну та безпечну роботу електронної пошти. Це передбачає мінімізацію як помилкових спрацювань (false positives), коли легітимні листи помилково позначаються як спам, так і пропусків (false negatives), коли спам потрапляє до вхідних скриньок. Для корпорацій, false positives можуть призвести до втрати важливих ділових комунікацій, тоді як false negatives підвищують ризик кіберінцидентів. Система повинна бути здатною ефективно обробляти великі та зростаючі обсяги пошти, забезпечуючи при цьому мінімальну затримку доставки. Фільтр повинен надавати надійний захист від широкого спектру кіберзагроз, включаючи фішинг, шкідливе програмне забезпечення, спуфінг та інші складні атаки, які постійно еволюціонують. Можливість безшовної інтеграції з існуючою

поштовою інфраструктурою організації (Mail Transfer Agents, поштові сервери, клієнти) є важливою для ефективного впровадження та управління.

Попри технічні можливості SpamAssassin, корпоративне розгортання стикається з низкою значних викликів. Спам постійно еволюціонує, щоб обходити фільтри, використовуючи нові методи та технології. Зокрема, зростання використання генеративних технологій штучного інтелекту (AI) дозволяє зловмисникам створювати більш переконливі атаки з імітацією особистості. Це включає імітацію стилів написання, термінологічних вподобань та комунікаційних шаблонів конкретних керівників або співробітників, що робить виявлення таких загроз надзвичайно складним для традиційних сигнатурних та правилових фільтрів. Це вимагає від корпоративних фільтрів не лише адаптивності, але й здатності до виявлення тонких, поведінкових аномалій. Безпека електронної пошти значною мірою залежить від взаємодії між технологіями та людськими аспектами. Більшість витоків даних (до 73.1%) спричинені ненавмисними людськими помилками, такими як неправильно адресовані листи, фішинг або недбалість. Це підкреслює, що навіть найдосконаліший технічний фільтр може бути скомпрометований через людську вразливість. Спамери експлуатують когнітивні упередження, використовуючи термінові повідомлення, авторитетні підказки та контент, що викликає страх, щоб обійти раціональне прийняття рішень у сфері безпеки. Навмисні дії зловмисників, які використовують привілейований доступ до поштових систем або знання організаційних вразливостей, становлять складні виклики для виявлення. Фрагментована інфраструктура безпеки, недостатнє навчання персоналу та надмірна залежність від стандартних платформ (наприклад, Microsoft 365) без належної конфігурації створюють "сліпі зони" та "дірки" в захисті. Багато організацій продовжують використовувати застарілі політики безпеки електронної пошти, які не враховують сучасні вектори загроз, такі як атаки з імітацією особистості. Крім того, відсутність належної підтримки кібербезпеки з боку керівництва може призвести до недостатнього виділення ресурсів та видимості для програм безпеки електронної пошти. Ці нетехнічні аспекти є такими ж, якщо не більш, критичними для безпеки електронної пошти, ніж суто технічні

вразливості. Недостатнє навчання співробітників, застарілі політики та фрагментована інфраструктура створюють прогалини в захисті, які не можуть бути повністю компенсовані лише технічними засобами фільтрації. Таким чином, для успішного корпоративного розгортання SpamAssassin необхідно інтегрувати його в ширшу стратегію безпеки, яка включає навчання користувачів, чіткі політики та регулярні перевірки.

Вибір архітектури розгортання SpamAssassin у корпоративному середовищі є стратегічним рішенням, що впливає на продуктивність, контроль, складність впровадження та загальну стійкість поштової системи. Існує кілька основних підходів до інтеграції SpamAssassin в інфраструктуру електронної пошти.

SpamAssassin може бути глибоко інтегрований з різними агентами передачі пошти (MTA), такими як Postfix, Sendmail, Exim та Qmail. Цей підхід дозволяє перевіряти пошту безпосередньо в потоці її обробки. Модель фільтрації до черги (Before-queue filtering) дозволяє SpamAssassin перевіряти повідомлення до того, як вони будуть повністю прийняті MTA та поміщені в чергу. Це є більш ефективним, оскільки дозволяє відхиляти спам на етапі SMTP-сесії, негайно повідомляючи відправника про відмову, що значно економить системні ресурси. Однак, така інтеграція є складнішою в налаштуванні. При фільтрації після черги (After-queue filtering) повідомлення спочатку приймаються MTA, а потім передаються SpamAssassin для перевірки. Це простіша в реалізації модель, але менш ефективна, оскільки спам вже прийнятий MTA, що споживає ресурси сервера. Для підвищення продуктивності у корпоративних середовищах рекомендується використовувати spamd (демон-версію SpamAssassin) та spamc (легкий клієнт), оскільки spamd працює постійно, усуваючи необхідність запускати повний Perl-інтерпретатор для кожного повідомлення, а spamc дозволяє MTA ефективно передавати повідомлення для перевірки. amavisd-new є високопродуктивним інтерфейсом, написаним на Perl, який слугує посередником між MTA та одним або кількома перевіряльниками вмісту (наприклад, вірусними сканерами та SpamAssassin), підтримуючи як фільтрацію до, так і після черги. milter-інтерфейс дозволяє інтегрувати SpamAssassin на рівні MTA, надаючи можливість відхиляти спам, повертаючи коди

помилок 5xx під час SMTP-сесії, що забезпечує раннє відхилення небажаної пошти. Хоча procmail є більш простим методом, він може бути використаний для інтеграції SpamAssassin із Sendmail для локальної доставки, дозволяючи використовувати spamd та задіювати файли користувацьких налаштувань та баєсівські бази даних. Пряма інтеграція SpamAssassin з Microsoft Exchange Server можлива через розробку спеціальних транспортних агентів, які приймають повідомлення, передають їх клієнту spamc для оцінювання, а потім, на основі отриманого балу, додають спеціальний заголовок, наприклад, X-Spam-Discard: YES, після чого Exchange Server може виконувати дії на основі цього заголовка.

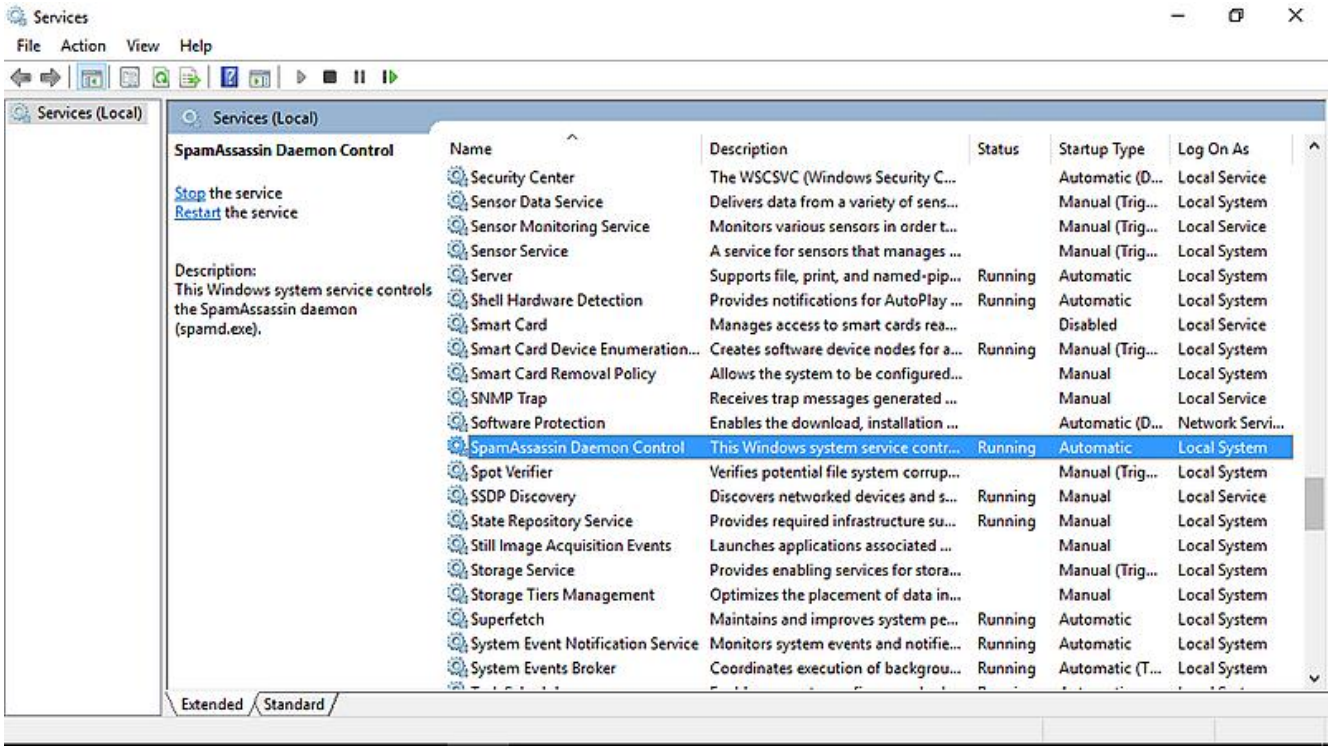


Рис.3.1. Інсталювання в Windows

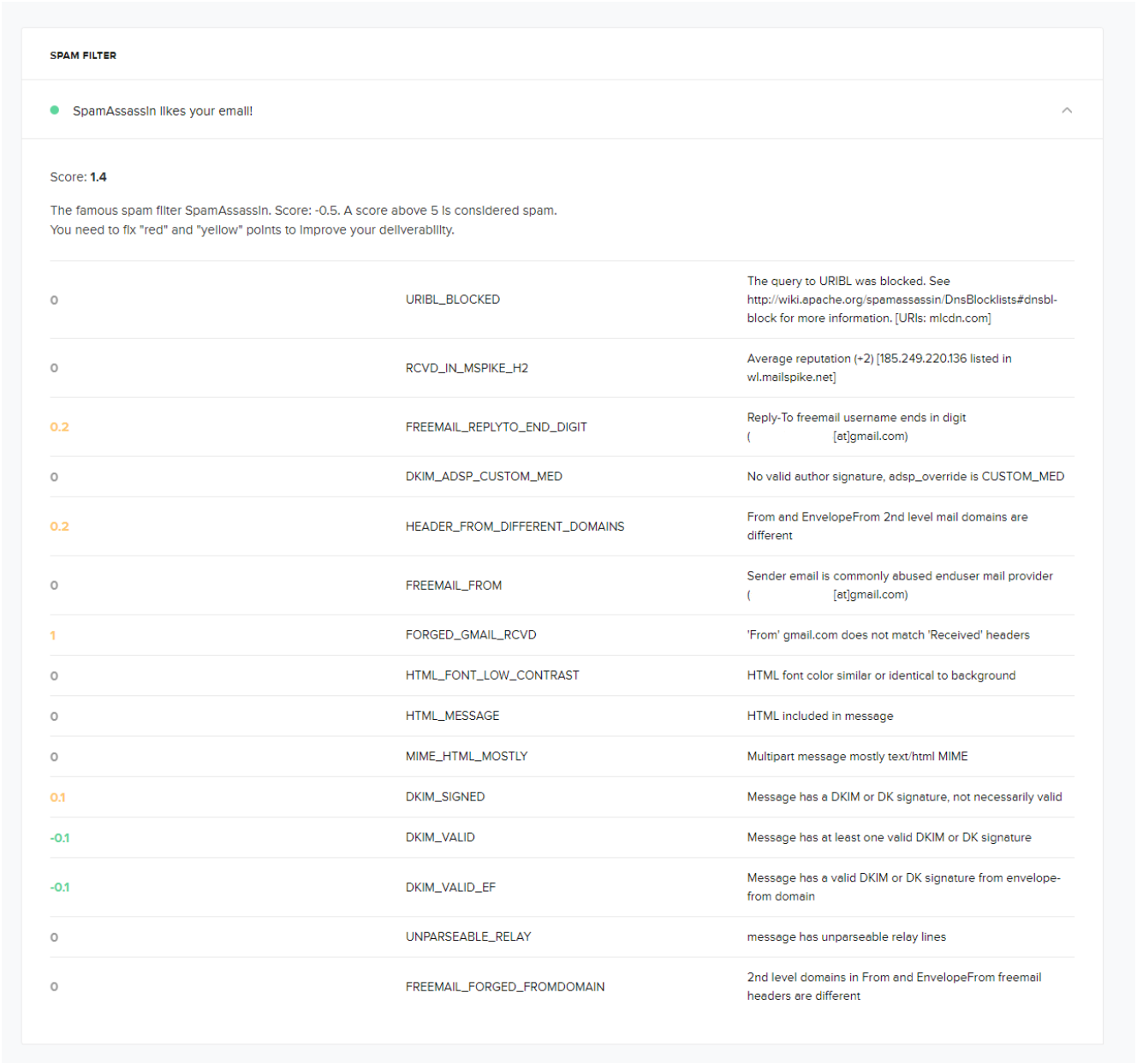
Альтернативним підходом є розгортання SpamAssassin як автономного шлюзу. У цій архітектурі SpamAssassin працює на окремій системі, яка розташована перед основним MTA організації. Вся вхідна пошта спочатку проходить через цей шлюз для фільтрації, а потім, якщо вона класифікується як легітимна, перенаправляється до внутрішнього MTA. Перевагами такого підходу є централізоване управління, що забезпечує єдину точку контролю для всіх правил фільтрації спаму, розвантаження MTA, оскільки основні поштові сервери

звільняються від інтенсивних обчислень, а також гнучкість інтеграції, що дозволяє легше інтегрувати автономний шлюз з іншими корпоративними системами, які потребують перевірки електронних листів, наприклад, маркетинговими системами для перевірки вихідних шаблонів. Зазвичай, автономний шлюз не відхиляє потенційний спам, а лише позначає його або переміщує до карантину, що запобігає втраті легітимних повідомлень через помилкові спрацювання. Недоліками є те, що окремий шлюз додає ще один компонент до поштової інфраструктури, який може стати єдиною точкою відмови, якщо не забезпечено високу доступність, а також може виникнути додаткова складність у налаштуванні маршрутизації пошти та DNS-записів (MX-записів) для перенаправлення трафіку через шлюз.

Вибір архітектури розгортання SpamAssassin у корпоративному середовищі є не просто технічним рішенням, а стратегічним, що впливає на продуктивність, контроль, складність впровадження та загальну стійкість поштової системи. Інтеграція до черги забезпечує раннє відхилення спаму, економлячи ресурси, але вимагає більшої складності конфігурації MTA. Автономний шлюз спрощує управління, але додає мережеву точку відмови. Рішення повинно ґрунтуватися на балансі між потребами в продуктивності (обсяг пошти), доступними IT-ресурсами (експертиза) та вимогами до безпеки (раннє відхилення).

Правильна конфігурація SpamAssassin є ключовою для забезпечення його ефективності, мінімізації помилкових спрацювань та пропусків, а також для підтримки стабільної роботи у корпоративному середовищі. Параметр required_score визначає мінімальний бал, який повинен набрати лист, щоб бути класифікованим як спам. Значення за замовчуванням 5.0 є досить агресивним і підходить для налаштування одного користувача, де допустимий вищий рівень помилкових спрацювань. Для корпоративного середовища або провайдерів рекомендується встановлювати більш консервативний поріг, наприклад, 8.0 або 10.0. Ця зміна допомагає значно зменшити кількість помилкових спрацювань (false positives), коли легітимні електронні листи неправильно позначаються як спам. Занадто агресивний поріг призводить до великої кількості false positives, що викликає скарги користувачів та знижує продуктивність. Надмірна кількість false

positives може підірвати довіру користувачів до системи та призвести до ручного обходу фільтрів, що, в свою чергу, знижує загальний рівень безпеки.



SPAM FILTER

SpamAssassin likes your email!

Score: **1.4**

The famous spam filter SpamAssassin. Score: -0.5. A score above 5 is considered spam.
You need to fix "red" and "yellow" points to improve your deliverability.

0	URIBL_BLOCKED	The query to URIBL was blocked. See http://wiki.apache.org/spamassassin/DnsBlocklists#dnsbl-block for more information. [URIs: mlcdn.com]
0	RCVD_IN_MSPIKE_H2	Average reputation (+2) [185.249.220.136 listed in wl.mailspike.net]
0.2	FREEMAIL_REPLYTO_END_DIGIT	Reply-To freemail username ends in digit ([at]gmail.com)
0	DKIM_ADSP_CUSTOM_MED	No valid author signature, adsp_override is CUSTOM_MED
0.2	HEADER_FROM_DIFFERENT_DOMAINS	From and EnvelopeFrom 2nd level mail domains are different
0	FREEMAIL_FROM	Sender email is commonly abused enduser mail provider ([at]gmail.com)
1	FORGED_GMAIL_RCVD	'From' gmail.com does not match 'Received' headers
0	HTML_FONT_LOW_CONTRAST	HTML font color similar or identical to background
0	HTML_MESSAGE	HTML included in message
0	MIME_HTML_MOSTLY	Multipart message mostly text/html MIME
0.1	DKIM_SIGNED	Message has a DKIM or DK signature, not necessarily valid
-0.1	DKIM_VALID	Message has at least one valid DKIM or DK signature
-0.1	DKIM_VALID_EF	Message has a valid DKIM or DK signature from envelope-from domain
0	UNPARSEABLE_RELAY	message has unparseable relay lines
0	FREEMAIL_FORGED_FROMDOMAIN	2nd level domains in From and EnvelopeFrom freemail headers are different

Рис.3.2. Класифікація листів як спам

Автоматичне видалення повідомлень, позначених як спам, не рекомендується. Користувачі можуть скаржитися на втрату легітимних листів. Якщо все ж необхідно автоматично видаляти повідомлення, це слід робити лише для листів з винятково високим балом, наприклад, 15.0 або вище, що свідчить про майже беззаперечний спам.

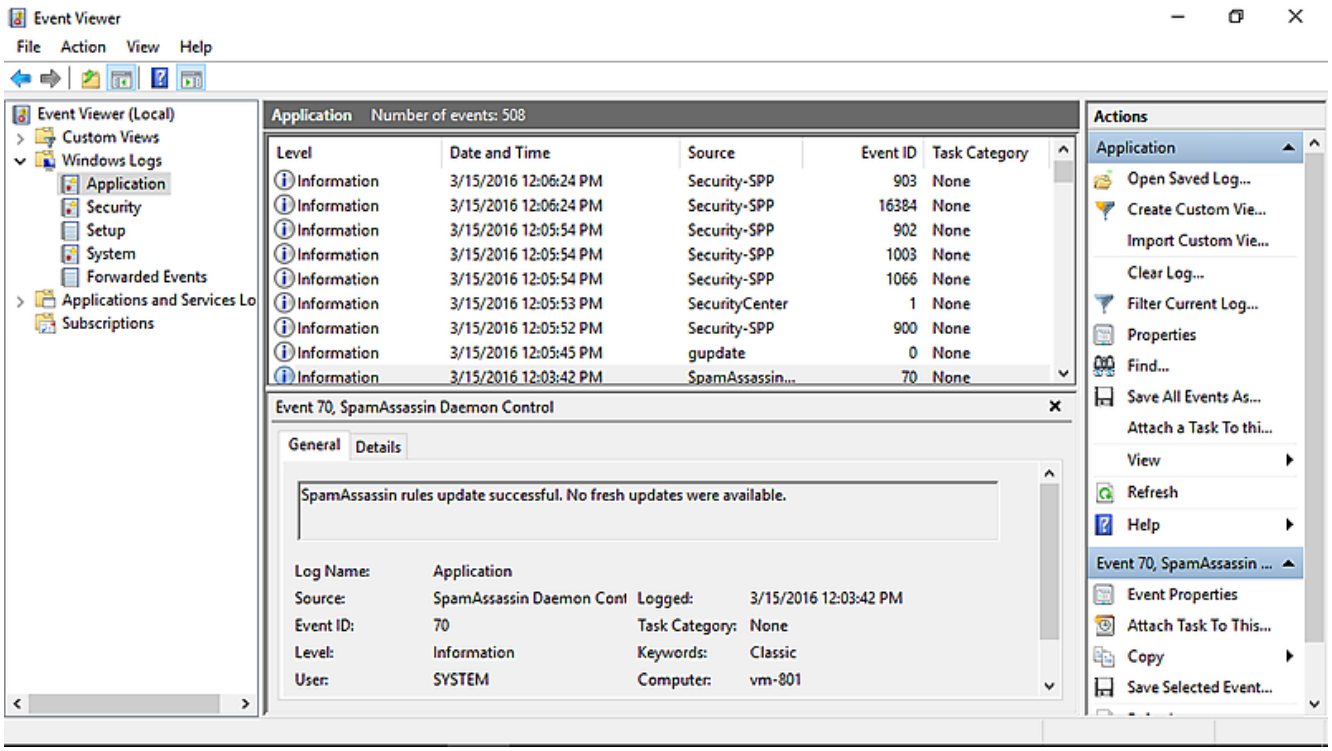


Рис.3.3. Логування в SpamAssassin

Ефективне управління правилами є життєво важливим для підтримки точності та безпеки SpamAssassin. Власні правила слід додавати до файлу /etc/mail/spamassassin/local.cf. Цей файл призначений для загальносистемних налаштувань і гарантує, що правила будуть застосовані незалежно від того, який користувач запускає SpamAssassin. Категорично не рекомендується додавати власні правила до файлів у /usr/share/spamassassin, оскільки вони будуть перезаписані під час оновлень SpamAssassin, що призведе до втрати всіх налаштувань. Крім того, увімкнення опції allow_user_rules, яка дозволяє окремим користувачам створювати власні правила у файлах user_prefs, не рекомендується для корпоративних середовищ. Це може мати значний вплив на навантаження сервера, оскільки SpamAssassin буде перекомпілювати всі тести при обробці повідомлення для користувача з власними правилами. Більш того, це створює потенційні проблеми безпеки: теоретично, зловмисний локальний користувач може експлуатувати spamd за допомогою хитромудрого регулярного виразу для отримання підвищених привілеїв. Безпека та продуктивність корпоративного розгортання SpamAssassin вимагають централізованого управління правилами, а не

дозволу користувачам створювати власні. Завжди перевіряйте синтаксис нових або змінених правил за допомогою команди spamassassin --lint. Для більш детального налагодження можна використовувати spamassassin --lint -D. Для ретельного тестування правил рекомендується використовувати власні або публічні корпуси (набори спаму та не-спаму) та інструменти, такі як mass_check, які дозволяють оцінити ефективність правил на великих обсягах даних. Для легкого розрізнення власних правил від стандартних, що постачаються з SpamAssassin, рекомендується додавати префікс, наприклад, LOCAL_ або ініціали адміністратора.

DNS-чорні списки (DNSBL) є важливим компонентом фільтрації спаму, оскільки вони дозволяють блокувати пошту від відомих джерел спаму. Для IP-адрес відправників рекомендується використовувати комбіновану зону Spamhaus Zen, яка включає SBL, XBL та PBL. Для доменів, що використовуються у спамі, рекомендується Spamhaus DBL. Ці списки забезпечують високий рівень виявлення спаму з низьким рівнем помилкових спрацювань. Як загальне правило, DNSBL, особливо PBL, не слід застосовувати до вихідної пошти організації, щоб уникнути помилкового блокування легітимних комунікацій. Можливість додавати власні RBL шляхом створення спеціальних конфігураційних файлів (наприклад, custom.cf) у каталозі /etc/mail/spamassassin/.

Ефективність SpamAssassin прямо залежить від безперервного оновлення правил та використання актуальних зовнішніх баз даних. Спамерські тактики постійно змінюються, тому застарілі правила швидко втрачають свою ефективність. Для забезпечення актуальності правил SpamAssassin необхідно щоденно запускати утиліту sa-update через планувальник завдань (cron). Важливо пам'ятати, що sa-update не перезапускає демон spamd автоматично після встановлення оновлених правил. Для застосування нових правил необхідно додати команду перезапуску spamd до скрипту оновлення (наприклад, sa-update && /etc/init.d/spamassassin reload). Це свідчить про те, що спам-фільтрація – це не статичне налаштування, а постійна "гонка озброєнь". Для корпорацій це означає, що автоматизація оновлень та інтеграція з надійними джерелами загроз є обов'язковою оперативною вимогою для підтримки ефективного захисту.

У корпоративному середовищі SpamAssassin повинен не лише ефективно фільтрувати спам, але й забезпечувати високу продуктивність, бути масштабованим для обробки великих обсягів пошти та мати високу доступність для безперебійної роботи. Для досягнення оптимальної продуктивності SpamAssassin необхідно ретельно налаштувати його конфігурацію та інтеграцію. Встановлення обмеження часу на обробку повідомлення за допомогою параметра time_limit (наприклад, 50 секунд для фільтрації до черги) захищає від надмірного споживання ресурсів на обробку вироджених або надзвичайно складних повідомлень, а також від DoS-атак. Як вже зазначалося, вимкнення allow_user_rules значно зменшує навантаження на сервер, оскільки усуває необхідність перекомпіляції правил для кожного користувача. Запуск локального DNS-сервера (наприклад, BIND) на тому ж хості, що й SpamAssassin, для кешування DNS-запитів значно підвищує ефективність мережевих тестів та запобігає блокуванню запитів з боку деяких DNSBL, які можуть інтерпретувати велику кількість запитів як зловмисну активність. Встановлення та інтеграція Pyzor та Razor надає SpamAssassin доступ до додаткових, корисних мережевих правил, що підвищує точність виявлення спаму. Точне визначення довірених та внутрішніх мереж за допомогою trusted_networks та internal_networks гарантує, що перевірки DNSBL не виконуються для легітимного внутрішнього трафіку, що покращує продуктивність та зменшує кількість помилкових спрацювань. Слід уникати використання великих або ресурсомістких наборів правил, таких як blacklist та blacklist-uri, оскільки вони можуть значно впливати на продуктивність та споживання пам'яті. Також важливо перевіряти власні правила на наявність погано написаних регулярних виразів, які можуть експоненційно споживати ресурси. Використання sa-compile (доступно для SA 3.2.x та новіших версій) може підвищити продуктивність, компілюючи правила у швидший формат. Файлова база даних Bayes, яка використовується за замовчуванням, може стати значним вузьким місцем для продуктивності та масштабованості у високонавантажених або кластерних середовищах. Вона схильна до проблем з блокуванням файлів ("bayes database lock error") при високому навантаженні, особливо якщо увімкнено autolearn або auto-expiry токенів.

Для вирішення цих проблем рекомендується перенести базу даних Bayes на більш надійне рішення, таке як SQL (MySQL, PostgreSQL) або Redis. Це дозволяє спільно використовувати базу даних між кількома хостами, значно зменшує проблеми з блокуванням та підвищує загальну пропускну здатність.

Для забезпечення безперебійної роботи поштової системи у корпоративному середовищі, SpamAssassin повинен бути розгорнутий з урахуванням високої доступності та масштабованості. Кластери високої доступності - це групи фізичних або віртуальних хостів, які діють як єдина система, забезпечуючи безперервну доступність послуг. Вони використовуються для балансування навантаження, резервного копіювання та відмовостійкості. Існують моделі кластерів Active/Passive, де один вузол активно обробляє запити, тоді як інші перебувають у режимі очікування і переймають навантаження лише у випадку відмови активного вузла, та Active/Active, де всі вузли в кластері активно обробляють запити, розподіляючи навантаження та забезпечуючи підвищену пропускну здатність та відмовостійкість. Для розподілу вхідного поштового трафіку між кількома серверами SpamAssassin (або серверами MTA з інтегрованим SpamAssassin) необхідно використовувати балансувальник навантаження (наприклад, HAProxy). Балансувальник навантаження направляє запити до доступних серверів, забезпечуючи оптимальне використання ресурсів та відмовостійкість. У кластерних розгортаннях, де SpamAssassin працює на кількох серверах, важливо забезпечити синхронізацію баєсівських баз даних між ними. Це можна зробити шляхом експорту бази даних з одного сервера та імпорту на інші. Однак, для високонавантажених середовищ більш ефективним є використання централізованої спільної бази даних (SQL або Redis) для Bayes, що усуває необхідність ручного експорту/імпорту та забезпечує автоматичну синхронізацію. Досягнення високої доступності та масштабованості SpamAssassin у корпоративному середовищі вимагає більше, ніж просто розгортання кількох інстансів. Це вимагає ретельного планування спільного стану та балансування навантаження. Без належного балансування навантаження та управління спільними ресурсами (наприклад, Bayes DB) кластер SpamAssassin не зможе ефективно

розподіляти навантаження або забезпечувати відмовостійкість. Це перетворює просте розгортання на складний інженерний проект, що вимагає інвестицій у додаткові компоненти інфраструктури та експертизу для їх налаштування та підтримки.

Ефективність системи фільтрації спаму вимірюється не лише кількістю заблокованого спаму, але й здатністю мінімізувати помилкові спрацювання (false positives) та пропуски (false negatives). У корпоративному середовищі це є спільною відповідальністю IT-адміністраторів та кінцевих користувачів. Зменшення кількості легітимних листів, помилково позначених як спам, є пріоритетом для будь-якої організації. Як зазначалося раніше, збільшення required_score (наприклад, до 8.0-10.0) є ефективним способом зменшити кількість легітимних листів, позначених як спам. Використання welcomelist_auth є кращим варіантом, оскільки він перевіряє автентифікацію відправника (SPF/DKIM), перш ніж довіряти листу. Це безпечніше, ніж welcomelist_from, який сліпо довіряє адресу відправника, що легко підробляється спамерами. Організації можуть додавати власні правила для білих списків внутрішніх адрес електронної пошти, щоб запобігти помилковим спрацюванням. SpamAssassin дозволяє писати власні правила, які ідентифікують не-спам-повідомлення та присвоюють їм негативні бали. Це може бути використано для корекції false positives, коли певні типи легітимних листів постійно помилково позначаються як спамом. Хоча сильні негативні бали слід використовувати з обережністю, їх вплив на безпеку менш серйозний, ніж високі позитивні бали для false positives.

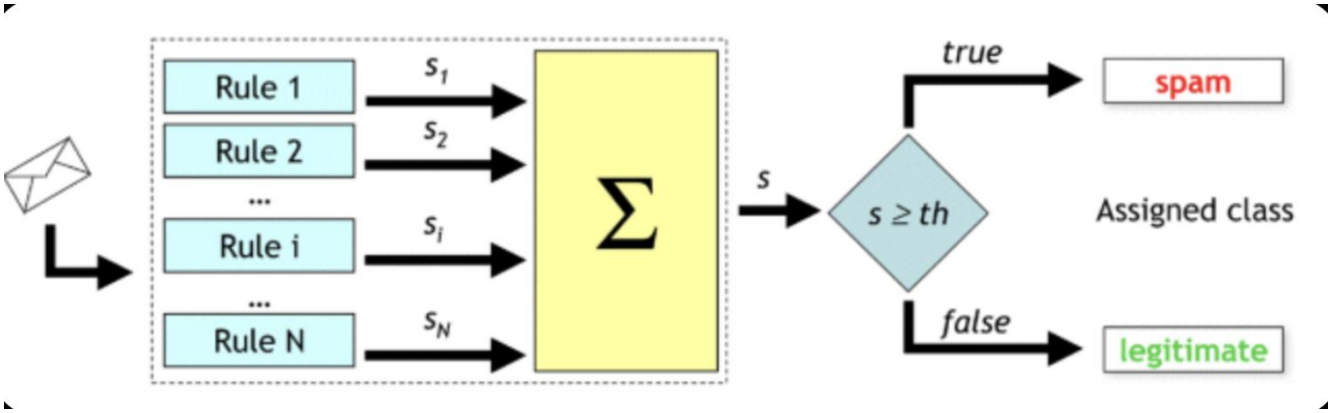


Рис.3.4. Створення правил в SpamAssassin

На рис.3.4. зображена концептуальна модель класифікації об'єктів, зокрема електронних повідомлень, що ґрунтується на агрегації оцінок, отриманих від множини дискретних правил. Зазначена модель відображає архітектурні принципи функціонування систем, подібних до SpamAssassin, щодо ідентифікації небажаної кореспонденції.

На початковому етапі відбувається надходження вхідного об'єкта – електронного повідомлення, що підлягає аналізу. Далі повідомлення піддається обробці численними правилами (від Rule 1 до Rule N), кожне з яких спрямоване на виявлення специфічних ознак або патернів. У контексті фільтрації спаму ці правила можуть охоплювати перевірку заголовків (наприклад, `Subject`, `From`), аналіз вмісту тіла повідомлення на наявність ключових слів або структурованих даних, оцінку URI-посилань, застосування мета-правил, що комбінують результати інших, а також проведення мережевих тестів, таких як запити до DNSBL або перевірки SPF/DKIM. Кожне правило, у разі його спрацювання, генерує відповідну індивідуальну оцінку (`s1, s2, ..., sN`), яка відображає ступінь його відповідності критеріям спаму.

Наступним етапом є агрегація всіх отриманих індивідуальних оцінок. Цей процес здійснюється шляхом підсумовування всіх балів, присвоєних спрацьованими правилами, що формує єдину сумарну оцінку `s` для аналізованого повідомлення.

Завершальний етап класифікації полягає у порівнянні отриманої сумарної оцінки `s` з попередньо встановленим пороговим значенням `th`. Якщо сумарна оцінка `s` перевищує або дорівнює пороговому значенню `th`, об'єкт класифікується як "спам". У протилежному випадку, тобто якщо `s` менше за `th`, об'єкт позначається як "легітимний". Таким чином, представлена модель демонструє застосування багатокритеріального підходу до визначення категорії об'єкта, де остаточне рішення приймається на основі кумулятивного ефекту численних індикаторів.

Зменшення кількості спаму, що потрапляє до вхідних скриньок, є не менш важливим для безпеки та продуктивності. Регулярне та послідовне навчання SpamAssassin на основі позначених користувачами спаму та не-спаму (ham) є критично важливим для підвищення точності баєсівського фільтра. Для ефективного навчання рекомендується використовувати мінімум 1000 повідомлень кожного типу (спам/ham), причому значне покращення точності спостерігається до 5000 повідомлень. Інтеграція з Pyzor, DCC та Vipul's Razor дозволяє SpamAssassin використовувати спільні бази даних загроз, що значно розширює його можливості виявлення спаму, який вже був ідентифікований іншими користувачами. Забезпечення щоденних оновлень правил SpamAssassin через sa-update є обов'язковим, оскільки спамерські тактики постійно змінюються, і застарілі правила швидко втрачають свою ефективність.

Корпорації повинні оптимізувати не лише вхідну фільтрацію, але й забезпечити, щоб їхні вихідні електронні листи не були помилково класифіковані як спам іншими системами. Недотримання найкращих практик для легітимних відправників призводить до того, що власні листи компанії потрапляють до спаму одержувачів, шкодячи діловій репутації та ефективності комунікацій. Налаштування протоколів автентифікації, таких як SPF (Sender Policy Framework), DKIM (DomainKeys Identified Mail) та DMARC, є фундаментальним для підтвердження легітимності відправника та запобігання спуфінгу. SpamAssassin перевіряє ці записи, і їх відсутність може негативно вплинути на бал. Підтримка позитивної репутації IP-адреси та домену є критично важливою, що досягається шляхом надсилання якісного, релевантного контенту, підтримки низьких показників відмов та високої взаємодії з одержувачами. Різкі сплески обсягу відправлення пошти або високі показники відмов можуть пошкодити репутацію. Слід уникати надмірного використання "спам-слів" (наприклад, "безкоштовно", "терміново", "гарантовано") та надмірної капіталізації у темі та тілі листа. Необхідно забезпечити якісне HTML-форматування без помилок, збалансоване співвідношення тексту до зображень (рекомендується 60% тексту та 40% зображень), а також завжди надавати альтернативний текст (alt text) для зображень.

Слід уникати прихованого тексту або невидимих веб-жучків. Також важливо бути обережним при включенні посилань на зовнішні домени, оскільки посилання на домени з поганою репутацією або ті, що перебувають у чорних списках, можуть негативно вплинути на репутацію відправника.

Ефективне управління false positives та false negatives є спільною відповідальністю IT-адміністраторів та кінцевих користувачів. Без активної участі користувачів у навчанні фільтра баєсівська база даних не буде точно відображати унікальні шаблони пошти організації, що призведе до зниження точності фільтрації. Необхідно забезпечити користувачам прості та інтуїтивно зрозумілі механізми для позначення листів як спаму або не-спаму (ham), що зазвичай реалізується через переміщення повідомлень до спеціальних папок ("Спам" або "Вхідні") у поштовому клієнті. Адміністратори можуть використовувати утиліту sa-learn для ручного навчання SpamAssassin на основі зібраних повідомлень, які були позначені користувачами, що дозволяє системі швидко адаптуватися до нових типів спаму та легітимних листів.

Забезпечення безпеки та ефективного моніторингу є невід'ємними частинами розгортання SpamAssassin у корпоративному середовищі, що дозволяє захистити систему від атак та підтримувати її оптимальну продуктивність. SpamAssassin, як і будь-яка система, що обробляє вхідний трафік, може бути мішенню для атак. Використання параметра time_limit у SpamAssassin є важливим заходом захисту від DoS-атак, оскільки він обмежує час, який SpamAssassin може витратити на обробку одного повідомлення, запобігаючи виснаженню ресурсів на "вироджених" або надзвичайно складних листах. Додатково, налаштування правил IDS/IPS (системи виявлення/запобігання вторгнень) на поштовому сервері може допомогти моніторити та блокувати підозрілу активність, таку як надмірна кількість з'єднань з однієї IP-адреси або ненормальна кількість повідомлень, надісланих за короткий проміжок часу. Категорично не рекомендується вмикати опцію allow_user_rules. Ця функція, хоча й забезпечує гнучкість для окремих користувачів, створює значний ризик ін'єкції коду та потенційного підвищення привілеїв, що може скомпрометувати весь поштовий сервер. Регулярне оновлення SpamAssassin є

критично важливим для виправлення відомих вразливостей (CVE), таких як ін'єкція коду в синтаксис мета-правил. Відкритість вихідного коду SpamAssassin сприяє швидкому виявленню та виправленню таких вразливостей спільнотою.

Для підвищення загальної безпеки розгортання SpamAssassin слід застосовувати наступні практики. Використання автентифікації SMTP для користувачів, що відправляють пошту, є надійною практикою безпеки, що усуває необхідність ведення білих списків динамічних IP-діапазонів та зменшує ризик несанкціонованої відправки пошти. Забезпечення правильних дозволів для файлів конфігурації SpamAssassin та баз даних є фундаментальним, оскільки файли повинні бути доступні лише для читання відповідним користувачам та групам, щоб запобігти несанкціонованим змінам. Періодичний аудит конфігураційних файлів та правил SpamAssassin допомагає виявити потенційні вразливості, неоптимальні налаштування або правила, які можуть бути використані зловмисниками.

Ефективний моніторинг дозволяє системним адміністраторам відстежувати продуктивність SpamAssassin, виявляти аномалії та реагувати на потенційні загрози. Кожне повідомлення, оброблене SpamAssassin, отримує додаткові заголовки, які містять детальну інформацію про його оцінку, включаючи X-Spam-Checker-Version, X-Spam-Level, X-Spam-Status та X-Spam-Report. Регулярний аналіз цих заголовків у вхідних повідомленнях є цінним інструментом для діагностики, розуміння причин оцінки та виявлення тенденцій. Для моніторингу ефективності SpamAssassin слід відстежувати загальну кількість оброблених повідомлень, кількість повідомлень, класифікованих як спам та не-спам (ham) , середній бал спаму та не-спаму, кількість помилкових спрацювань (false positives) та пропусків (false negatives), час обробки повідомлень , а також кількість з'єднань та відхилених повідомлень на MTA. Хоча SpamAssassin не має вбудованого експортера метрик для Prometheus, його лог-файли та вихідні дані можуть бути використані для інтеграції з комплексними системами моніторингу, такими як Nagios, Prometheus та Splunk. Наприклад, Nagios може моніторити SMTP-сервіси та генерувати сповіщення на основі визначених параметрів та порогів. Інформація про кількість спаму/не-спаму, час обробки та інші статистичні дані може бути

вилучена з лог-файлів SpamAssassin (наприклад, за допомогою sa-stats.pl) та імпортована до систем моніторингу для візуалізації та аналізу. Деякі поштові сервери (наприклад, James) надають метрики SpamAssassin TCP client statistics, які можуть бути експортовані для Prometheus. Rspamd, альтернативний спам-фільтр, має вбудований експортер метрик для Prometheus. Дані, що генеруються SpamAssassin (наприклад, модифіковані заголовки, записи в логах), можуть бути інтегровані в SIEM (Security Information and Event Management) системи, що дозволяє корелювати події, пов'язані зі спамом, з іншими подіями безпеки в корпоративній мережі, забезпечуючи більш повний огляд стану кібербезпеки.

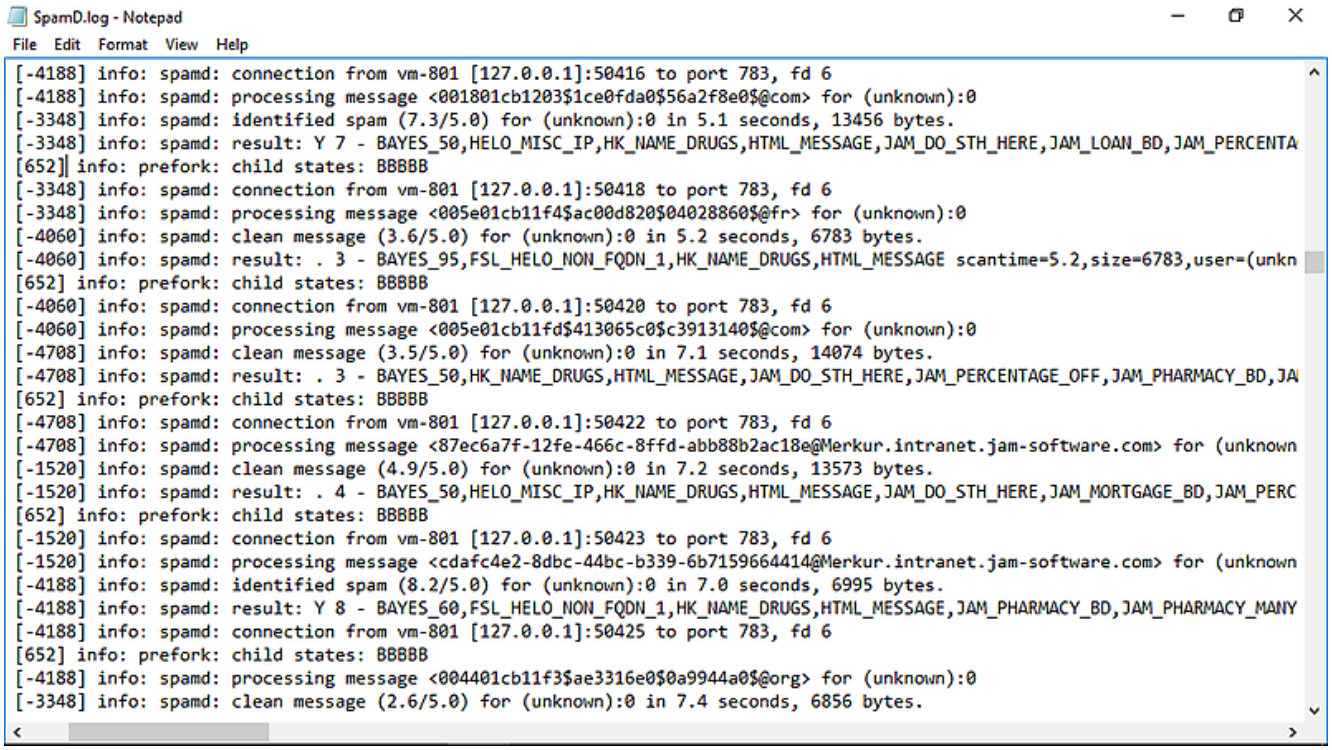


Рис.3.5. Заблоковані спам листи

Лог-файли SpamAssassin містять цінну інформацію для налагодження та оптимізації. SpamAssassin може генерувати детальні лог-повідомлення, включаючи інформацію про спрацьовані правила, бали та час обробки. Утиліта sa-stats.pl є прикладом скрипта, який може аналізувати лог-файли SpamAssassin для збору статистики щодо спаму та не-спаму, середніх балів, часу обробки та обсягу даних. Це дозволяє адміністраторам отримувати уявлення про ефективність фільтра та виявляти аномалії. Дослідження також показують, що баєсівські спам-фільтри

можуть бути ефективно використані для класифікації та фільтрації записів у лог-файлах загалом, що підкреслює потенціал для розширеного аналізу [10].

3.2. Розробка рекомендацій щодо виявлення фішингових повідомлень засобами SpamAssassin

Розгортання Apache SpamAssassin у корпоративному середовищі є складним, але надзвичайно важливим завданням, що вимагає глибокого розуміння як технічних аспектів, так і організаційних та людських факторів. Ефективна фільтрація спаму вийшла за рамки простої зручності, ставши критичним компонентом стратегії кібербезпеки та підтримки бізнес-продуктивності.

Ключові висновки свідчать, що спам більше не є лише "шумом"; він перетворився на складний вектор кібератак, що використовує фішинг, шкідливе програмне забезпечення та навіть AI-генеровані імітації особистості. Це вимагає від SpamAssassin не лише адаптивності, але й здатності до виявлення тонких, поведінкових аномалій, а також постійного оновлення правил та інтеграції з актуальними джерелами загроз. Ефективність SpamAssassin ґрунтується на його багатоспектральному підході, що поєднує евристичний аналіз, баєсівську фільтрацію, DNSBL та спільні бази даних. Жоден окремий метод не є достатнім, тому необхідно забезпечити належне функціонування всіх компонентів. Технічні рішення, якими є SpamAssassin, не можуть повністю компенсувати ризики, пов'язані з людськими помилками, недостатнім навчанням, застарілими політиками та фрагментованою інфраструктурою. Інтеграція SpamAssassin має бути частиною ширшої стратегії безпеки, що включає навчання користувачів та чіткі політики. Стратегічний вибір архітектури між інтеграцією з MTA та автономним шлюзом впливає на продуктивність, контроль та складність. Автономний шлюз може бути кращим для великих, складних середовищ, а також для перевірки вихідних маркетингових листів. Файлова баєсівська база даних є значним вузьким місцем

для продуктивності та масштабованості. Для високонавантажених або кластерних розгортань необхідно перенести її на SQL або Redis та ретельно планувати балансування навантаження та управління спільними ресурсами. Ефективне управління помилковими спрацюваннями та пропусками вимагає активної участі кінцевих користувачів у навчанні баєсівського фільтра. Корпорації також повинні оптимізувати свої вихідні електронні листи, щоб уникнути помилкової класифікації як спаму іншими системами.

На основі цих висновків, надаються наступні рекомендації. Для оптимізації конфігурації слід встановити консервативний поріг required_score (8.0-10.0) для зменшення помилкових спрацювань, але уникати автоматичного видалення листів з низьким балом. Рекомендується використовувати welcomelist_auth для білих списків та розглянути правила з негативними балами для корекції false positives. Завжди слід додавати власні правила до /etc/mail/spamassassin/local.cf та уникати allow_user_rules через безпекові ризики та вплив на продуктивність. Необхідно регулярно перевіряти правила за допомогою spamassassin --lint та тестувати їх на корпусах.

Для підвищення продуктивності та масштабованості слід встановити time_limit (наприклад, 50 секунд) для захисту від DoS-атак та надмірної обробки. Рекомендується розгорнути локальний кешуючий DNS-сервер для підвищення ефективності мережевих тестів. Баєсівську базу даних слід перенести на централізовану SQL або Redis для кластерних розгортань та високого навантаження. Для ефективної інтеграції з MTA слід використовувати spamd/spamc та amavisd-new.

Для забезпечення високої доступності необхідно планувати розгортання SpamAssassin у кластерній архітектурі (Active/Active або Active/Passive) з використанням балансувальника навантаження. Важливо забезпечити синхронізацію або спільне використання баєсівських баз даних між вузлами кластера.

Безперервне оновлення та навчання є обов'язковими. Слід автоматизувати щоденні оновлення правил SpamAssassin за допомогою sa-update та забезпечити

автоматичний перезапуск spamd. Користувачів слід заохочувати активно позначати спам та не-спам для постійного навчання баєсівського фільтра. Рекомендується використовувати DNSBL, такі як Spamhaus Zen та DBL.

Для комплексної безпеки необхідно забезпечити належну автентифікацію вихідної пошти (SPF, DKIM, DMARC) та підтримувати високу репутацію відправника. Важливо навчати користувачів розпізнавати фішингові та AI-генеровані атаки, оскільки людський фактор залишається критичною вразливістю. Слід регулярно оновлювати SpamAssassin для виправлення відомих вразливостей та застосовувати практики загартування системи.

Нарешті, для моніторингу та аналізу необхідно впровадити моніторинг ключових метрик SpamAssassin (кількість спаму/не-спаму, бали, час обробки) через аналіз заголовків та лог-файлів. Слід розглянути інтеграцію даних SpamAssassin з централізованими системами моніторингу (Nagios, Prometheus, Splunk) та SIEM для агрегованого аналізу та швидкого реагування на інциденти.

Впровадження цих рекомендацій дозволить корпораціям створити надійну, масштабовану та безпечну систему фільтрації спаму на базі Apache SpamAssassin, ефективно протидіючи сучасним кіберзагрозам та забезпечуючи безперебійну роботу електронної пошти [11].

3.3. Рекомендації щодо підвищення ефективності фільтрації фішингу в корпоративній поштовій системі

Фішингові атаки залишаються одним із найбільш розповсюджених типів соціально-інженерних загроз у корпоративному середовищі. Вони характеризуються використанням електронної пошти для введення користувачів в оману з метою отримання конфіденційної інформації, такої як облікові дані, фінансові реквізити або доступ до внутрішніх ресурсів підприємства. У зв’язку з цим ефективна фільтрація фішингових повідомлень є пріоритетним завданням у сфері забезпечення інформаційної безпеки організацій.

Підвищення ефективності фільтрації фішингових повідомлень потребує впровадження багаторівневих систем захисту, що поєднують декілька механізмів перевірки. Сучасні підходи базуються на використанні алгоритмів машинного навчання для аналізу структурних і семантичних характеристик повідомлень. Такі алгоритми дозволяють виявляти закономірності у форматуванні листів, використовуваній лексиці, метаданих заголовків і шаблонах вкладених файлів. Додатково впровадження поведінкового аналізу взаємодії з поштовими сервісами дозволяє виявляти аномальні дії, які можуть свідчити про спробу доставки фішингового контенту.

Ефективність фільтрації значною мірою залежить від регулярного оновлення баз сигнатур і моделей класифікації. Оскільки фішингові кампанії постійно еволюціонують, системи фільтрації мають отримувати актуальні дані про нові загрози в режимі реального часу. Це досягається шляхом інтеграції з зовнішніми джерелами кіберрозвідки, глобальними репутаційними сервісами, системами обміну індикаторами компрометації (IoC), а також через зворотний зв’язок із результатами внутрішнього моніторингу інцидентів.

Ключовим аспектом технічного захисту електронної пошти є впровадження протоколів автентифікації, таких як SPF (Sender Policy Framework), DKIM (DomainKeys Identified Mail) і DMARC (Domain-based Message Authentication, Reporting & Conformance). Ці протоколи забезпечують перевірку автентичності джерела електронного листа, дозволяють відслідковувати спроби підміни адреси відправника та забезпечують централізовану політику реагування на підозрілі повідомлення [12].

Не менш важливим чинником забезпечення захищеності поштової інфраструктури є підвищення обізнаності співробітників щодо фішингових загроз. Проведення регулярних навчань, симуляцій атак і тестування навичок розпізнавання фішингу дозволяє зменшити ймовірність того, що користувач стане жертвою соціальної інженерії. Впровадження внутрішніх політик кібергігієни, включаючи регламент перевірки джерел повідомлень, заборону переходу за

підозрілими посиланнями та використання двофакторної автентифікації, також підвищує загальну стійкість організації до фішингу.

Додатково, системи електронної пошти мають забезпечувати можливість карантинування підозрілих повідомлень. Такий підхід дозволяє запобігти безпосередньому контакту користувача з потенційно шкідливим контентом, одночасно надаючи адміністраторам можливість здійснити ручну або автоматизовану перевірку повідомлення. Інструменти журналювання та моніторингу подій повинні бути інтегровані в загальну систему інформаційної безпеки організації, що дозволяє оперативно реагувати на інциденти та проводити їхній ретроспективний аналіз із метою вдосконалення політик фільтрації.

Таким чином, забезпечення ефективної фільтрації фішингових повідомлень у корпоративній поштовій системі вимагає комплексного підходу, що поєднує технологічні рішення, управлінські стратегії та заходи з підвищення обізнаності персоналу. Лише скоординована робота усіх компонентів системи дозволяє створити стійке інформаційне середовище, здатне протистояти сучасним загрозам соціальної інженерії.

ВИСНОВКИ

У кваліфікаційній роботі розглянуто проблему протидії фішинговим атакам в корпоративному середовищі електронної пошти. Встановлено, що фішинг залишається одним із найбільш розповсюджених і небезпечних типів соціально-інженерних атак, спрямованих на викрадення конфіденційної інформації, компрометацію облікових записів користувачів та порушення цілісності корпоративної інформаційної системи. Актуальність проблеми обумовлена зростаючою складністю фішингових кампаній, використанням технік обходу фільтрів та широким залученням соціальних маніпуляцій.

Проведений аналіз дозволив виділити сучасні методи виявлення та блокування фішингових повідомлень, серед яких ключове місце займають системи контентної фільтрації, машинне навчання, протоколи автентифікації пошти (SPF, DKIM, DMARC), а також спеціалізовані поштові шлюзи. Зокрема, увагу приділено рішенню на базі відкритого програмного забезпечення Apache SpamAssassin, яке є одним із найпоширеніших інструментів для фільтрації небажаної пошти, включаючи фішинг.

У межах кваліфікаційної роботи проведено налаштування та тестування системи захисту на базі SpamAssassin в умовах корпоративної поштової інфраструктури. SpamAssassin реалізує багатоетапний підхід до аналізу поштових повідомлень, поєднуючи евристичні правила, байєсівську фільтрацію, DNSBL-перевірки, аналіз заголовків і вмісту листів, а також можливість інтеграції зі сторонніми сервісами й модулями машинного навчання. Однією з ключових переваг є гнучкість у налаштуванні правил та можливість адаптації до специфіки організації.

Було проаналізовано архітектуру рішення SpamAssassin, яка дозволяє застосовувати фільтрацію в реальному часі та інтегруватися з іншими елементами поштової інфраструктури, зокрема Postfix, Dovecot та Amavis.

Проведене тестування показало, що після налаштування системи із урахуванням локальних патернів фішингу та типових загроз, рівень детекції фішингових листів значно підвищився. Також оцінено вплив налаштувань на продуктивність поштового сервера та якість фільтрації (відношення істинно-позитивних до хибно-позитивних спрацювань).

Окремо розглянуто можливості автоматичного оновлення правил, взаємодії з репутаційними сервісами та побудови власних сигнатур на основі попередньо виявлених інцидентів. Показано, що адаптивний механізм SpamAssassin дає змогу не лише знижувати кількість фішингових повідомлень, які досягають користувача, але й підвищує загальну ефективність поштової безпеки завдяки гнучкому механізму керування ризиками.

Під час реалізації рішення запропоновано методологію інтеграції SpamAssassin у корпоративне середовище з урахуванням наявних обмежень і технічної специфікації мережі. Визначено доцільність комбінації SpamAssassin із системами DMARC-політик, а також впровадження додаткових модулів автоматичного реагування на підозрілі листи.

У результаті впровадження засобів захисту на базі SpamAssassin було досягнуто покращення ефективності виявлення фішингу, мінімізації кількості інцидентів соціальної інженерії та зниження ризиків витоку інформації. Рекомендовано використовувати даний інструмент як частину багаторівневої системи захисту електронної пошти підприємства в поєднанні з навчанням персоналу, ретельним моніторингом поштової активності та систематичним оновленням фільтраційних політик. Таким чином, впровадження рішення на базі SpamAssassin дозволяє створити ефективний бар'єр проти фішингових атак в умовах корпоративного середовища.

ПЕРЕЛІК ПОСИЛАНЬ

1. AI Phishing in 2025: Emerging Threats & Defense Strategies. *Sasa Software*. URL: https://www.sasa-software.com/blog/ai-phishing-attacks-defense-strategies/ (дата звернення: 14.03.2025).
2. PEEK: Phishing Evolution Framework for Phishing Generation and Evolving Pattern Analysis using Large Language Models. *arXiv.org e-Print archive*. URL: https://arxiv.org/html/2411.11389v2 (дата звернення: 18.03.2025).
3. 8 Phishing Techniques. *Check Point*. URL: https://www.checkpoint.com/cyber-hub/threat-prevention/what-is-phishing/8-phishing-techniques/ (дата звернення: 20.03.2025).
4. 10 Top Tips to Detect Phishing Scams. *SecurityHQ*. URL: https://www.securityhq.com/blog/top-tips-to-detect-phishing-scams/ (дата звернення: 25.03.2025).
5. 5 Real-Time Techniques to Detect Phishing Attack. *LoginRadius*. URL: https://www.loginradius.com/blog/identity/real-time-techniques-detect-phishing-attacks (дата звернення: 02.04.2025).
6. Spam Assassin as a Service. *Hassen Taidirt*. URL: https://www.hassen.io/posts/2024/10/spam-assassin-as-a-service/ (дата звернення: 02.04.2025).
7. SpamAssassin Spam Filter. *Documentation and Help Portal for Plesk Obsidian*. URL: https://docs.plesk.com/en-US/obsidian/administrator-guide/mail/antispam-tools/spamassassin-spam-filter.59432/ (дата звернення: 04.04.2025).
8. Mail::SpamAssassin::Conf - SpamAssassin configuration file. *Apache SpamAssassin: Welcome*. URL: https://spamassassin.apache.org/full/3.3.x/doc/Mail_SpamAssassin_Conf.html (дата звернення: 09.04.2025).
9. Apache SpamAssassin 4.0.0 Release: Enhanced Classification. *Makes Email Safe For Business: Microsoft 365, Google Workspace.* URL: https://guardiandigital.com/resources/blog/apache-spamassassin-4-0-0-released (дата звернення: 11.04.2025)

10. 14.4.5 SpamAssassin Configuration Examples (Sun Java System Messaging Server 6.3 Administration Guide). *Moved*. URL: https://docs.oracle.com/cd/E19566-01/819-4428/bgate/index.html (дата звернення: 11.04.2025).

11. Rajashree. What is SpamAssassin Score and How to Fix It. *Cold Email Outreach Tool | Premium Deliverability Designed for Outbound*. URL: https://www.smartlead.ai/blog/what-is-spamassassin-score-and-how-to-fix-it (дата звернення: 16.04.2025).

12. SpamAssassin Mail Filter. *Webmin*. URL: https://webmin.com/docs/modules/spamassassin-mail-filter/ (дата звернення: 18.04.2025).