ризику, розширення бази політик безпеки та застосування машинного навчання для виявлення аномалій у конфігураціях.

Перелік посилань

1. Center for Internet Security. CIS Docker Benchmark v1.6.0 [Електронний ресурс] // CIS. 2023. Режим доступу: https://www.cisecurity.org/benchmark/docker.

2. Docker. Docker Bench for Security [Електронний ресурс] // GitHub Repository. 2024. Режим доступу: https://github.com/docker/docker-bench-security.

3. Aqua Security. Trivy: A Simple and Comprehensive Vulnerability Scanner for Containers and IaC [Електронний ресурс] // Aqua Security. 2024. Режим доступу: https://aquasec.com/tools/trivy.

4. Bridgecrew. Checkov: Infrastructure as Code static analysis tool [Електронний ресурс] // Bridgecrew. 2024. Режим доступу: https://www.checkov.io/.

5. Checkmarx. KICS – Keeping Infrastructure as Code Secure [Електронний ресурс] // Checkmarx Docs. 2024. Режим доступу: https://docs.kics.io/latest/queries/dockercompose-queries.

6. GoodwithTech. Dockle – Container Image Linter for Security [Електронний ресурс] // GitHub Repository. 2023. Режим доступу: https://github.com/goodwithtech/dockle.

7. OWASP Foundation. Container Security Verification Standard (CSVS), Draft v0.9 [Електронний ресурс] // OWASP. 2023. Режим доступу: https://owasp.org/.

8. Docker. Managing Secrets in Docker Compose [Електронний ресурс] // Official Documentation. 2024. Режим доступу: https://docs.docker.com/compose/use-secrets/.

9. OWASP Foundation. Docker Security Cheat Sheet [Електронний ресурс] // OWASP Cheat Sheet Series. 2023. Режим доступу: https://cheatsheetseries.owasp.org/cheatsheets/Docker_Security_Cheat_Sheet.html.

10. Docker Documentation. Networking in Compose [Електронний ресурс] // Docker Docs. 2024. Режим доступу: https://docs.docker.com/compose/networking/.

11. Docker Documentation. Compose Deploy Resources [Електронний ресурс] // Docker Docs. 2024. Режим доступу: https://docs.docker.com/compose/compose-file/deploy/#resources.

12. Snyk Ltd. Container Security: Volume Mount Risks [Електронний ресурс] // Snyk Learn. 2023. Режим доступу: https://snyk.io/learn/docker-security/.

13. Red Hat. Running Containers as Non-root [Електронний ресурс] // Red Hat Container Security. 2022. Режим доступу: https://www.redhat.com/en/topics/containers.

14. National Institute of Standards and Technology (NIST). Application Container Security Guide (SP 800-190) [Електронний ресурс] // NIST Publications. 2017. Режим доступу: https: // nvlpubs. nist. gov / nistpubs / SpecialPublications/NIST.SP.800-190.pdf.

15. Open Policy Agent. Rego Policy Language Documentation [Електронний ресурс] // OPA Docs. 2024. Режим доступу: https://www.openpolicyagent.org/docs/latest/.

16. GitLab Documentation. Shift Left Security in CI/CD Pipelines [Електронний ресурс] // GitLab Docs. 2023. Режим доступу: https://docs.gitlab.com/ee/topics/security/shift_left_security.html.

17. Docker Documentation. Compose File Reference (v3.9) [Електронний ресурс] // Docker Docs. 2024. Режим доступу: https://docs.docker.com/compose/compose-file/.

18. Google Cloud Security. Machine Learning for Misconfiguration Detection [Електронний ресурс] // Google Cloud Security Blog. 2023. Режим доступу: https://cloud.google.com/security.

Надійшла 10.11.2025

УДК 004.056.5:004.7
DOI: 10.31673/2409-7292.2025.041215

Прокопович-Ткаченко Д.І., Зубченко Н.С., Черкаський О.В., Черкаський Д.О., Тихоненко І.М.

МЕТОДИ ВИЯВЛЕННЯ ШКІДЛИВИХ ВЛАСТИВОСТЕЙ АЛЬТЕРНАТИВНИХ ПРОШИВОК МАРШРУТИЗАТОРІВ У КОНТЕКСТІ СУЧАСНИХ МЕРЕЖЕВИХ ЗАГРОЗ

У статті досліджується використання альтернативних та модифікованих прошивок маршрутизаторів як одного з ключових чинників формування ботнет-мереж і реалізації розподілених DDoS-атак. Показано, що підроблені або змінені прошивки можуть містити приховані бінарні модулі, алгоритми псевдовипадкової генерації доменів, схеми відкладеної активації та елементи архітектури block-safe, що забезпечують стійкі та малопомітні канали C2-взаємодії. Проведений аналіз трафіку, порівняння контрольних сум прошивок та часткова зворотна інженерія виявили типові ознаки компрометації: періодичні cron-виклики, звернення до нестандартних доменних зон, використання нетипових TCP/UDP-портів, а також активацію шкідливого функціоналу у відповідь

на структурно фіксовані пакет-тригери. Отримані результати демонструють, що маршрутизатори з такими прошивками здатні непомітно включатися до ботнет-інфраструктури, передавати телеметрію на C2-сервери, брати участь у короткочасних, але інтенсивних DDoS-кампаніях і після цього повертатися у фоновий режим без залишення помітних артефактів. Запропонований алгоритм виявлення включає інтеграцію NetFlow-моніторингу, SIEM-кореляції та перевірки цілісності прошивок, що дозволяє ідентифікувати приховані механізми взаємодії навіть за умов мінімальної мережевої активності. Результати роботи формують підґрунтя для вдосконалення підходів до виявлення шкідливих прошивок і підвищення стійкості мереж до сучасних IoT-ботнетів, а також окреслюють перспективи створення поведінково орієнтованих firmware-aware систем захисту.

Ключові слова: альтернативні прошивки, ботнет-мережі, DDoS-атаки, приховані C2-канали, block-safe архітектура, пакет-тригери, NetFlow-моніторинг, SIEM-кореляція, IoT-безпека, зворотна інженерія, компрометація маршрутизаторів.

Вступ

Стрімке зростання інцидентів інформаційної безпеки, пов’язаних із використанням мережевого обладнання як платформи для формування ботнет-мереж, стало однією з ключових тенденцій останнього десятиліття. Маршрутизатори, точки доступу, камери спостереження та інші пристрої Інтернету речей дедалі частіше використовуються зловмисниками як низькозахищені вузли для прихованого розгортання шкідливих компонентів, що підтверджується сучасними дослідженнями у сфері IoT-безпеки [1–5]. Масовість такого обладнання, відсутність своєчасних оновлень безпеки та недостатній рівень контролю з боку користувачів створюють сприятливі умови для здійснення масштабних DDoS-кампаній та формування стійких ботнет-інфраструктур [6–10]. У межах цієї проблематики особливе місце займають модифіковані або підроблені прошивки маршрутизаторів. Вони містять приховані модулі керування, бекдори, механізми псевдовипадкової генерації доменів, схеми відкладеної активації та системи маскування шкідливої взаємодії. Такі прошивки приховують трафік за легітимними DNS-запитами або фоновими системними процесами, що значно ускладнює виявлення їхньої активності навіть у випадку застосування сучасних IDS і поведінкових систем аналізу [11–14].

Найскладніші екземпляри реалізують архітектуру block safe, у якій шкідливі сигнали шифруються, розподіляються між множиною вузлів і активуються лише під час надходження специфічних сигнатурних пакетів [15–17]. Попри наявність великої кількості робіт, присвячених виявленню ботнетів, більшість досліджень зосереджені на аналізі мережевого трафіку або застосуванні алгоритмів машинного навчання. Натомість питання, пов’язані зі структурою альтернативних прошивок, способами інтеграції прихованих C2 каналів, специфічними патернами активації та ризиками для критичної інфраструктури, висвітлені неповно [18–20].

Це створює прогалину між сучасними методами аналізу шкідливого трафіку та реальними сценаріями проникнення, у яких джерелом компрометації виступає саме прошивка, а не класичне зараження. У цьому дослідженні ставиться мета комплексно проаналізувати альтернативні прошивки маршрутизаторів, які містять приховані модулі керування, реалізують архітектуру block safe та здатні включати обладнання до ботнет-мереж [21].

Особлива увага приділяється виявленню їхніх архітектурних ознак, характерних патернів трафіку, внутрішніх механізмів взаємодії та ризиків, що виникають як для домашніх, так і для корпоративних і критичних інфраструктур. Результатом роботи є систематизований опис поведінкових характеристик таких прошивок, визначення ознак компрометації, опис механізмів C2 взаємодії та побудова алгоритму виявлення модифікованих прошивок за допомогою NetFlow моніторингу, аналізу контрольних сум та кореляції подій у SIEM системах. Сформовано комплекс рекомендацій щодо політики управління ризиками, централізованого менеджменту оновлень, сертифікації дозволених образів та процедур аудиту обладнання. Отримані результати створюють наукове й практичне підґрунтя для вдосконалення механізмів раннього виявлення та нейтралізації ботнет-активності на рівні прошивок маршрутизаторів.

Постановка проблеми

Масове використання мережевого обладнання та IoT-пристроїв призвело до зростання інцидентів, у яких саме модифіковані або підроблені прошивки маршрутизаторів виступають первинним джерелом компрометації. На відміну від класичних шкідливих програм, такі прошивки інтегрують шкідливі модулі безпосередньо у файлову структуру пристрою, забезпечуючи приховану активність, стійкі C2-канали, DGA-алгоритми та механізми відкладеної активації. Їхня робота малопомітна для традиційних засобів IDS/IPS, що фокусуються на аналізі трафіку, а не на поведінці та структурі firmware.

Попри значну кількість досліджень, більшість з них обмежуються аналізом мережевих ознак IoT-ботнетів, тоді як архітектурні та поведінкові характеристики шкідливих прошивок досліджені недостатньо. Невирішеними залишаються питання виявлення прихованих модулів, ідентифікації пакет-тригерів, перевірки цілісності прошивок і реконструкції логіки взаємодії з C2-інфраструктурою. Це створює критичну прогалину між реальною динамікою firmware-орієнтованих атак та можливостями наявних засобів моніторингу.

Таким чином, актуальною є потреба у комплексному підході, що поєднує аналіз трафіку, контроль цілісності прошивок та часткову зворотну інженерію, дозволяючи виявляти прошивки з прихованими модулями, block-safe елементами та шкідливими схемами взаємодії. Лише інтегрований метод здатний забезпечити раннє виявлення та нейтралізацію firmware-орієнтованих ботнет-загроз у сучасних мережах.

Аналіз останніх публікацій

У сучасних дослідженнях IoT-безпеки значна увага приділяється виявленню ботнет-мереж, особливо тих, що використовують маршрутизатори та інше мережеве обладнання як базову інфраструктуру. Роботи зосереджуються переважно на мережевому поведінковому аналізі: класифікації трафіку, використанні гібридних ML-моделей, оптимізації ознак та побудові детекторів для IoT-середовищ. Публікації також демонструють активний розвиток методів аналізу DDoS-кампаній, моделей фільтрації та оцінювання ризику використання скомпрометованих пристроїв.

Водночас окрема група досліджень вивчає структуру та функції шкідливих прошивок, зокрема використання прихованих модулів, sandbox-аналіз, відкладені механізми активації та нетипові патерни DNS-/TCP-/UDP-трафіку. Деякі автори описують інтеграцію block-safe схем, DGA-алгоритмів та FSM-подібних тригерів, що ускладнюють детектування. Досліджено також питання зворотної інженерії та ідентифікації змінених бінарних компонентів.

Попри значний прогрес, більшість робіт розглядає лише окремі аспекти проблеми – або мережеву поведінку ботнетів, або статичний аналіз firmware. Комплексних підходів, що одночасно враховують структурні, динамічні та поведінкові ознаки модифікованих прошивок, все ще бракує. Саме ця прогалина обумовлює необхідність подальших досліджень, спрямованих на інтеграцію методів аналізу трафіку, перевірки цілісності та реконструкції логіки шкідливих модулів.

Мета і завдання дослідження

Метою дослідження є встановлення того, як альтернативні та модифіковані прошивки маршрутизаторів можуть містити приховані шкідливі модулі, формувати стійкі C2-канали й використовуватися у ботнет-інфраструктурах. Дослідження передбачає аналіз структури таких прошивок, виявлення їхніх поведінкових та мережевих аномалій, визначення механізмів активації шкідливого функціоналу й розроблення узагальненого підходу до виявлення прихованої компрометації, що дозволяє оцінити ризики їхнього використання в сучасних мережах.

Методологія дослідження

Методологія дослідження сформована з урахуванням специфіки загроз, пов’язаних із використанням альтернативних модифікованих прошивок маршрутизаторів, які здатні приховано встановлювати командно-керувальні канали, взаємодіяти через схеми block-safe та

забезпечувати тривалу непомічену активність. На відміну від класичних шкідливих програм, приховані модулі вбудовуються безпосередньо у файлову структуру прошивки, тому їхнє виявлення потребує комплексного підходу: аналізу мережевого трафіку, кореляції подій у NetFlow/IPFIX, порівняння контрольних сум та зворотної інженерії бінарних компонентів.

Обрана методологія включає три взаємодоповнюючі напрями:

1. Динамічний аналіз мережевого трафіку;
2. Аналіз цілісності прошивок через порівняння контрольних сум;
3. Часткову зворотну інженерію підозрілих модулів та реконструкцію їхньої логіки.

Нижче наведено повний формалізований опис підрозділу, присвяченого динамічному аналізу трафіку.

Еволюція ботнетів на базі мережевого обладнання

Перші покоління ботнетів базувалися переважно на інфікуванні персональних комп'ютерів та традиційних операційних систем, використовуючи класичні вектори атак: фішинг, експлойти браузера та заражені вкладення. Із поширенням мобільних пристроїв та IoT акцент змістився на масове компрометування мережевого обладнання, зокрема маршрутизаторів, що мають дефолтні паролі, відкриті порти віддаленого адміністрування та вразливі веб-інтерфейси. Дослідження також фіксують активне використання слабко захищених прошивок як точки проникнення у ботнет-мережі, особливо тих, що піддаються модифікуванню або заміні сторонніми образами [11]. Розвиток ботнетів засвідчує еволюцію від простих централізованих C2-архітектур до складних розподілених структур із використанням алгоритмів динамічної генерації доменів (DGA), які ускладнюють блокування командних серверів. Моделювання доменних імен, характерних для IoT-ботнетів, демонструє, що зловмисники активно використовують низьку керованість побутових маршрутизаторів для довготривалої прихованої активності у мережі [12].

Паралельно посилюється тенденція використання IoT-пристроїв у DDoS-кампаніях, що спричиняє необхідність дослідження тактик протидії атакам на канальному рівні. Зокрема, моделі оптимального розподілу фільтрів пропонують знижувати навантаження на вузли, які можуть бути використані ботнетами як трафік-генератори [13]. Серед ключових характеристик ботнетів нового покоління актуальні наукові роботи виділяють: масовість потенційних жертв, слабкий контроль стану пристроїв користувачами та довгі періоди непоміченої діяльності. Ці чинники роблять маршрутизатори та інше мережеве обладнання привабливою платформою для створення прихованих шкідливих екосистем у межах IoT-середовища.

Альтернативні прошивки: легітимний і тіньовий сегменти

Альтернативні прошивки, такі як OpenWrt або DD-WRT, спочатку створювалися для розширення функцій обладнання, підтримки додаткових протоколів, VPN-рішень або QoS-механізмів. Вони мають відкритий код, що знижує ризик прихованих шкідливих модулів. Однак зростання популярності таких рішень спричинило появу тіньового сегмента — модифікованих образів прошивок, створених для інтеграції бекдорів, прихованих модулів керування та інших шкідливих компонентів. Дослідження виявляють, що такі прошивки можуть приховано збирати інформацію про пристрій, відкривати нестандартні порти, виконувати циклічні фонові запити та ініціювати підключення до шкідливої інфраструктури. Порівняння бінарних модулів й агрегація інформативних ознак за допомогою машинного навчання дозволяють ідентифікувати прошивки, що зазнали несанкціонованої модифікації [14]. У складніших випадках тіньові прошивки вбудовують елементи глибинного навчання або системи автоматичного вибору тактик взаємодії, що ускладнює виявлення, особливо в умовах великих розподілених інфраструктур IoT. У деяких роботах описано використання гібридних моделей машинного навчання, які дозволяють виявляти поведінкові аномалії навіть у модифікованих системних компонентах [15]. Крім того, запропоновано економічні стратегії відновлення IoT-інфраструктур після атак, які передбачають очищення або оновлення

прошивок, що знижують ризики тривалого збереження прихованих шкідливих компонентів у мережі [16]. Значну увагу науковці приділяють аналізу шкідливих прошивок у контрольованих умовах. Використання sandbox-середовищ дозволяє досліджувати поведінку модулів у реальному часі та виявляти приховані системні виклики, які не проявляються в умовах статичного аналізу [17].

Схема block-safe та приховані C2-канали

Архітектура block-safe описує тип поведінкової взаємодії, за якої шкідливі командно-керувальні сигнали маскуються під легітимний мережевий трафік, а маршрутизація команд проходить через низку проміжних вузлів. Такі прошивки використовують техніки псевдовипадкової генерації доменів, періодичні шифровані запити та приховані «сплячі» модулі, що активуються лише після отримання спеціальних тригерів. У роботах, присвячених гібридним метаевристичним моделям аналізу IoT-ботнетів, зазначається, що алгоритми такого типу значно ускладнюють виявлення ботнет-трафіку в реальному часі [18].

Особливо важливим є той факт, що деякі прошивки здатні динамічно змінювати поведінку в залежності від отриманих команд. До прикладу, у системах smart-міст зафіксовано використання SAE-GRU моделей для моделювання трафіку, які одночасно служать основою для прихованих каналів зв'язку у модифікованих прошивках [19]. Сучасні підходи на основі нечітких нейтрософічних множин демонструють, що block-safe інфраструктури формують складні патерни трафіку, які практично неможливо виявити за допомогою сигнатурного аналізу. Такі системи здатні змінювати інтенсивність та структуру передачі даних, залишаючись непоміченими в умовах періодичного моніторингу [20]. Таким чином, наявні дослідження підтверджують, що приховані C2-канали, вбудовані в підроблені або модифіковані прошивки, становлять один з найбільш складних і маловивчених класів загроз для сучасних мережевих інфраструктур.

Динамічний аналіз трафіку

Для виявлення ознак прихованих C2-каналів застосовано динамічний аналіз мережевої активності на рівні pcap-трейсів, NetFlow/IPFIX-записів та журналів firewall/IDS. Метою аналізу є побудова формалізованого профілю нормальної поведінки маршрутизатора та виділення аномалій, що можуть бути пов'язані зі «сплячими» або активними шкідливими компонентами прошивки.

Нехай для інтервалу спостереження T формується множина потоків $F_k,\ k = 1, \dots, K$. Кожен потік описується вектором ознак:

$$\mathbf{x}_k = (V_k, N_k, P_k, D_k), \tag{1}$$

де V_k – загальний обсяг байтів, N_k – кількість пакетів, P_k – порт призначення, D_k – домен/адреса призначення.

Сумарний обсяг трафіку у вікні Δt задається:

$$X(t_i) = \sum_{k \in \mathcal{F}(t_i)} V_k, \qquad t_i = i \cdot \Delta t. \tag{2}$$

Оцінки середнього та стандартного відхилення:

$$\mu_X = \frac{1}{N}\sum_{i=1}^{N} X(t_i), \qquad \sigma_X = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N} (X(t_i) - \mu_X)^2}. \tag{3}$$

Аномальні інтервали:

$$z_X(t_i) = \frac{X(t_i) - \mu_X}{\sigma_X}, \qquad |z_X(t_i)| > \theta_X. \tag{4}$$

Аналогічно формується часовий ряд запитів до доменів другого/третього рівня:

$$C_{d_j}(t_i) = \text{кількістьзапитівдо}d_j\text{у}[t_i, t_i + \Delta t],$$

$$z_{d_j}(t_i) = \frac{C_{d_j}(t_i) - \mu_{d_j}}{\sigma_{d_j}}. \quad (5)$$

Ознакою прихованого C2 є стабільна періодичність DNS/HTTP-запитів, що визначається автокореляційною функцією:

$$R_{d_j}(\tau) = \frac{1}{N-\tau}\sum_{i=1}^{N-\tau} (C_{d_j}(t_i) - \mu_{d_j})(C_{d_j}(t_{i+\tau}) - \mu_{d_j}). \quad (6)$$

Інтегральна аномальна оцінка:

$$S(t_i) = \alpha|z_X(t_i)| + \beta \max_j |z_{d_j}(t_i)|. \quad (7)$$

MATLAB-візуалізації та реалізація аналізу

Для аналізу поведінки маршрутизаторів із альтернативними прошивками було побудовано часовий профіль мережевого трафіку, що дозволяє виявляти різкі відхилення від типового фону та фіксувати можливі ознаки активації прихованих C2-модулів. Такий підхід дає змогу порівняти нормальну низькорівневу активність пристрою з нетиповими піками, які можуть бути пов'язані з виконанням шкідливих команд.

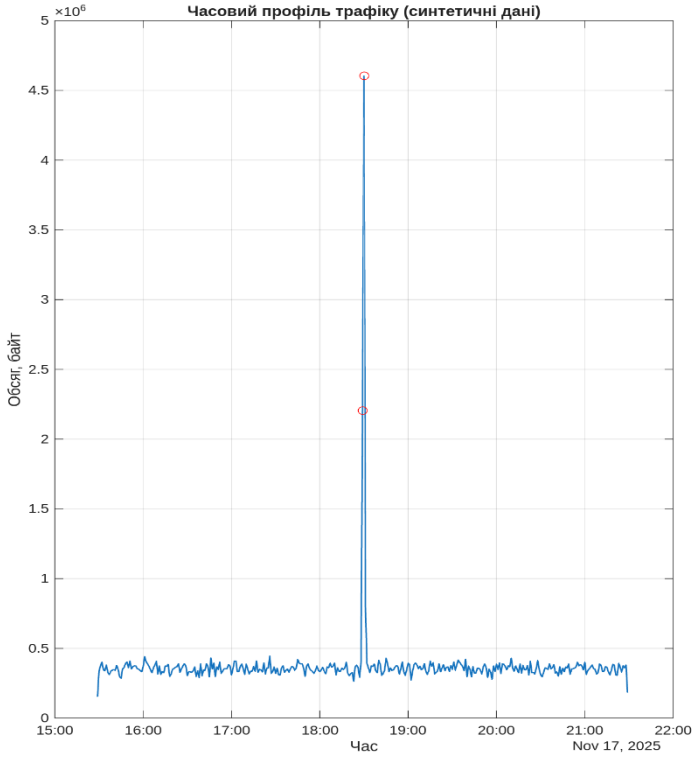

Рис. 1. Часовий профіль трафіку

Рисунок 1 демонструє часовий профіль трафіку, у якому більшу частину періоду спостереження обсяг переданих даних залишається стабільним і коливається в межах типового фону маршрутизатора. Близько 18:00 відбувається різкий стрибок трафіку обсягом понад 4,6×10 байт, що значно перевищує нормальну активність. Такий різко виражений пік свідчить про нетипову подію, яка не відповідає звичайній роботі прошивки. Подібні короткочасні сплески часто пов'язані з активацією прихованих модулів, ініціацією сеансу зв'язку з командно-керувальною інфраструктурою або виконанням короткострокових шкідливих операцій, що характерні для модифікованих прошивок з ознаками ботнет-

поведінки. Часовий портовий розподіл трафіку було використано для виявлення нетипових або прихованих шаблонів взаємодії, які не завжди можна визначити за загальним обсягом переданих даних. Такий підхід дозволяє оцінити, які порти використовуються пристроєм упродовж часу, та визначити можливі відхилення у вигляді нестандартних або періодичних звернень, характерних для модулів прихованого керування. Аналіз цієї залежності дає змогу виявляти аномальні мережеві поведінки, що можуть бути пов'язані з альтернативними або модифікованими прошивками маршрутизаторів.

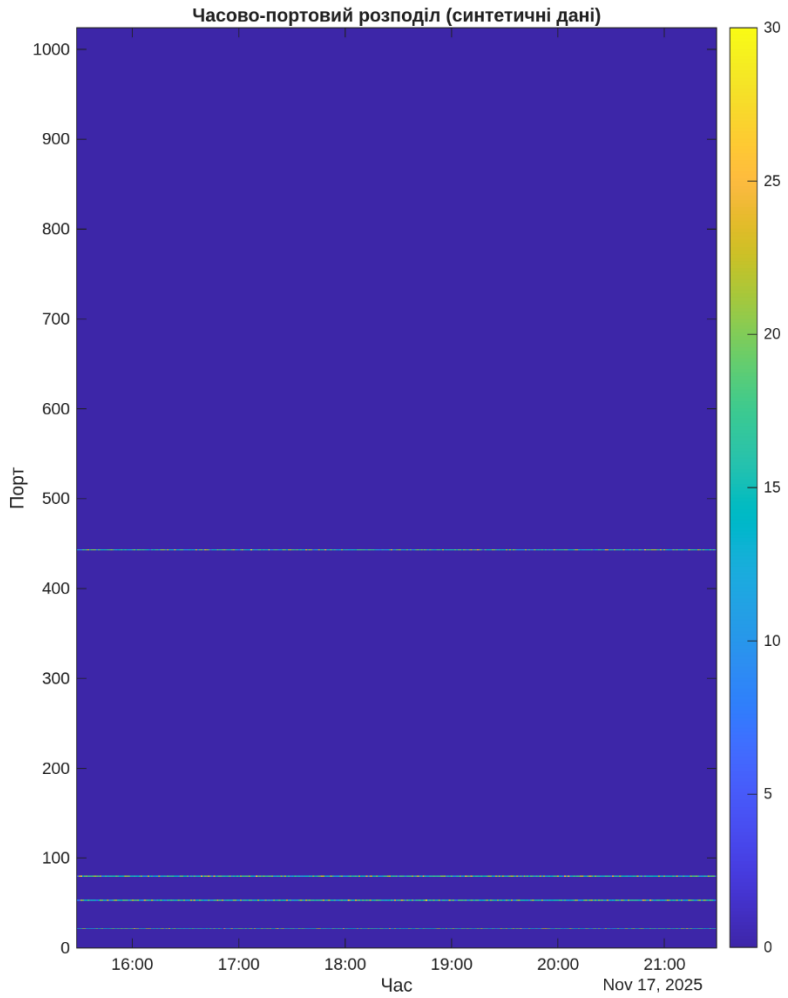

Рис. 2. Теплова карта розподілу використання портів

Рисунок 2. Показує теплову карту розподілу використання портів протягом усього періоду спостереження.Основна частина трафіку генерується в діапазонах низьких портів, де видно кілька горизонтальних ліній, що відповідають стабільно використовуваним сервісам. Інтенсивність їхнього використання мінімальна, що характерно для фонових системних процесів. Окремі порти періодично активні у невеликому діапазоні від 50 до 120, а також в області біля 450 порту.

Аналіз активності окремих доменів використовується для виявлення нетипових або прихованих регулярних звернень, які можуть свідчити про роботу шкідливих модулів, маскування C2-взаємодії під легітимний DNS-трафік або аномальні періодичні запити. Для маршрутизаторів із модифікованими прошивками характерними є стабільні повторювані звернення до певних доменів, що суттєво відрізняються від нормальної поведінки. Побудова часової серії DNS-активності дозволяє виявити такі патерни й оцінити, наскільки звернення

до конкретних доменів відповідають типовому користувацькому навантаженню. Рисунок 3 відображає динаміку кількості звернень до домену google.com на основі синтетичних даних.

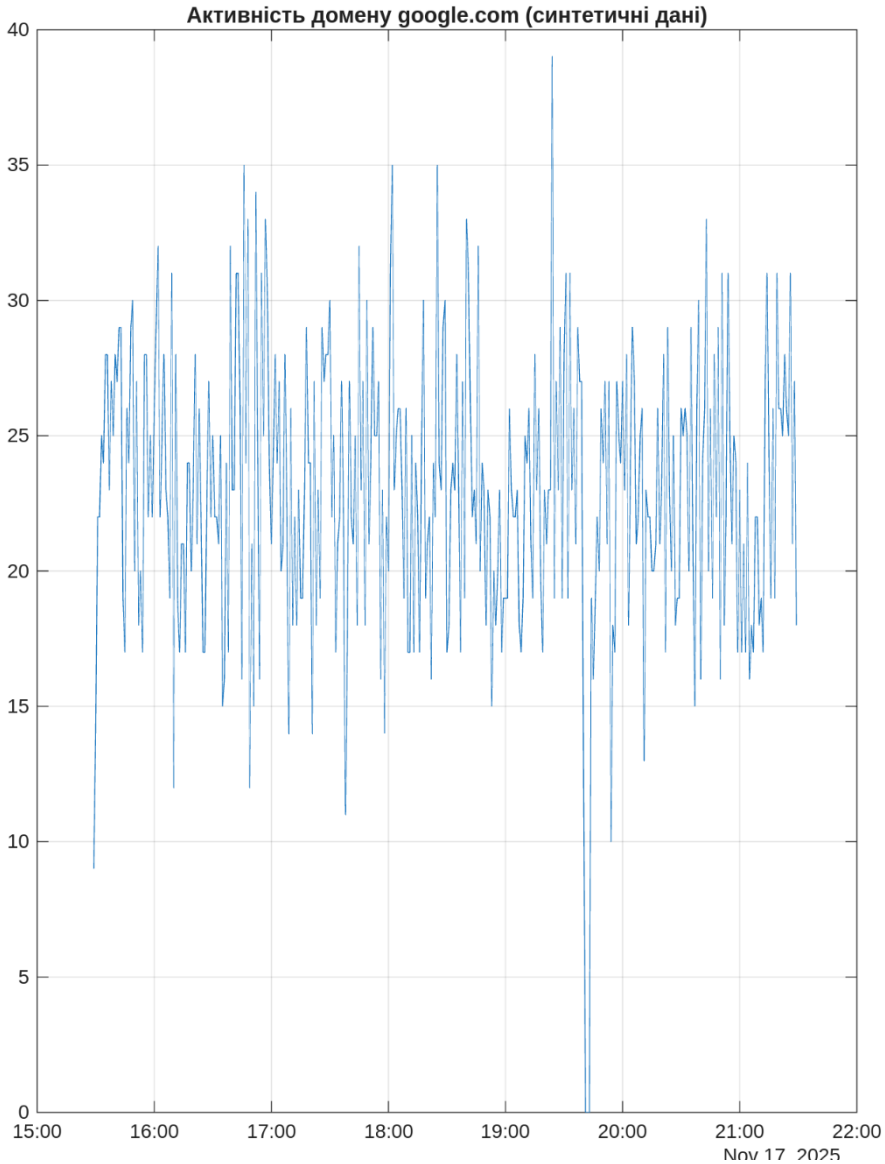

Рис. 3. Динаміка кількості звернень до домену google.com

Упродовж усього періоду спостереження значення коливаються в межах від приблизно 15 до 35 запитів за інтервал, що є характерним для активного, але звичайного використання домену, наприклад під час фонового оновлення сервісів або роботи браузера. Часовий ряд має значну варіативність, але не демонструє чіткої періодичності або регулярних циклів, які могли б вказувати на приховану системну взаємодію чи діяльність шкідливих модулів. Наявність поодиноких падінь до нульових значень пояснюється короткими інтервалами без активності, що теж не виходить за межі нормальної поведінки. Загальна картина свідчить про те, що активність домену є випадковою та рівномірно розподіленою, без ознак стабільних синхронізованих запитів, характерних для прихованих C2-каналів у змінених прошивках.

Автокореляційний аналіз використовується для визначення прихованої періодичності у часових рядах DNS-запитів, що дозволяє виявити регулярні цикли, характерні для прихованих каналів керування у модифікованих прошивках. Якщо у зверненнях до певного домену присутня повторювана часово зсунутина структура, автокореляційна функція покаже

виражені піки на певних лагах. У випадку стандартної користувацької активності часовий ряд є випадковим і не має стабільної періодичності, тому значення автокореляції залишаються низькими за винятком нульового лага.

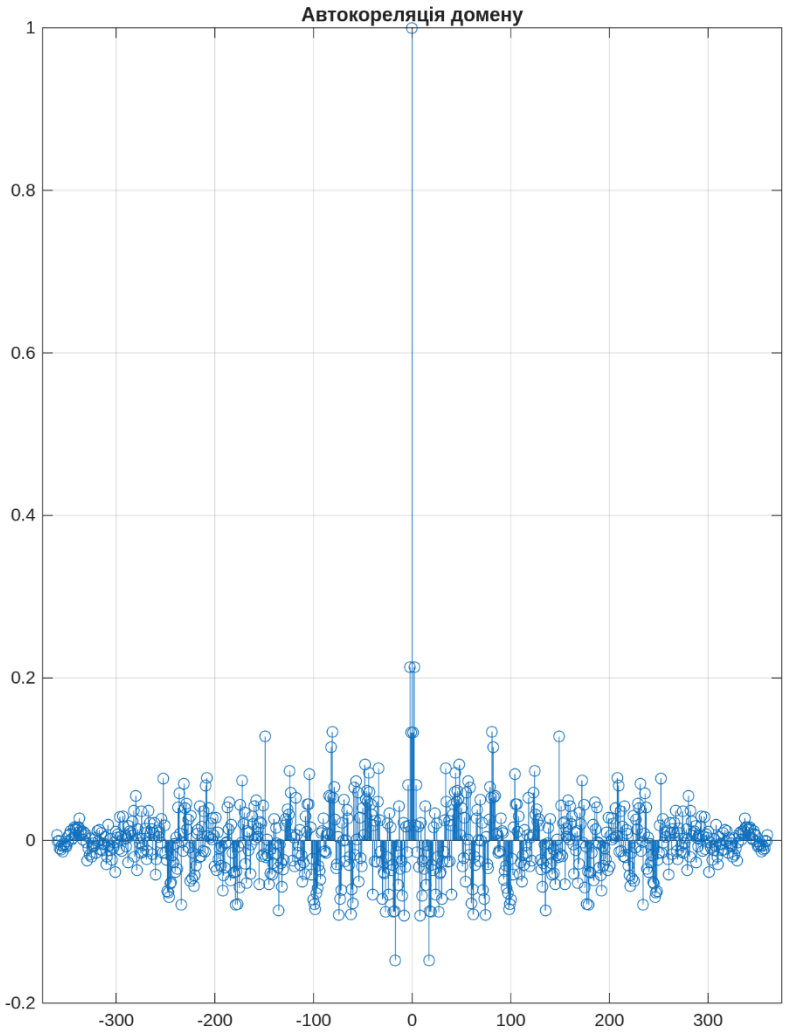

Рис. 4. Результат автокореляції активності домену

Рисунок 4 відображає результати автокореляції активності домену. Головний пік на нульовому зсуві є очікуваним і відповідає повній кореляції сигналу із самим собою. Усі інші значення коливаються навколо нуля і не формують виражених повторюваних піків на жодному з лагів. Такий вигляд автокореляційної функції свідчить про відсутність регулярних часових патернів у зверненнях до аналізованого домену. Часовий ряд не демонструє циклічності чи стабільної повторюваної структури, що характерно для нормальної фонової активності. Водночас така картина не вказує на наявність прихованих C2-механізмів, оскільки шкідливі модулі зазвичай створюють чіткі повторювані інтервали, які формують додаткові кореляційні піки на певних часових зсувах.

Інтегральна аномальна оцінка S(t) використовується для узагальнення різних показників мережевої активності в єдину метрику, що дозволяє швидко визначати моменти часу, коли поведінка пристрою істотно відхиляється від норми. Такий підхід поєднує властивості обсягу трафіку, періодичності звернень, активності портів та інших характеристик, формуючи сумарний індикатор аномальності. Його застосування особливо ефективне при виявленні короткочасних, але значущих проявів прихованих модулів у модифікованих прошивках, які часто не помітні при аналізі окремих параметрів.

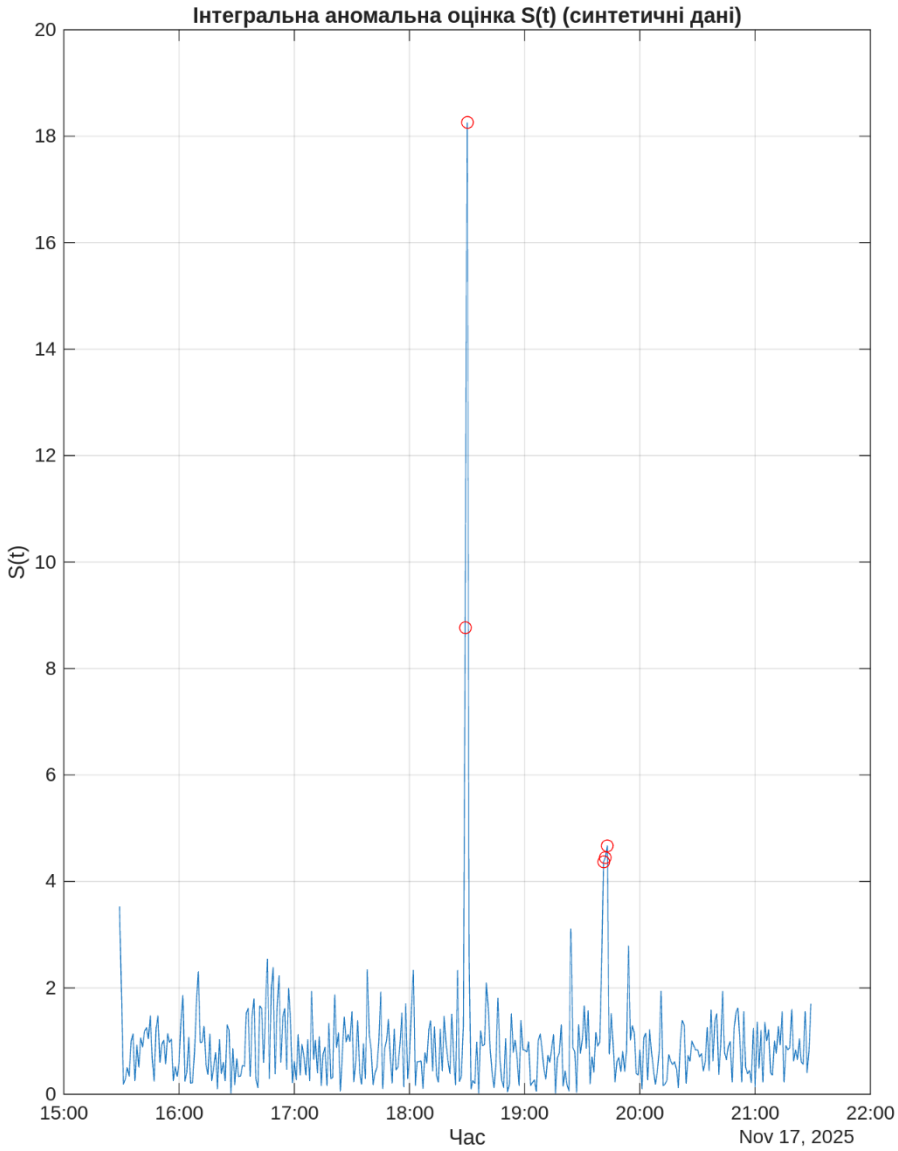

Рис. 5. Часова залежність інтегральної оцінки S(t) для синтетичних даних

Рисунок 5 демонструє часову залежність інтегральної оцінки S(t) для синтетичних даних. Більшу частину часу показник залишається у межах низьких значень, що відповідає нормальній фоновій активності маршрутизатора. Проте приблизно о 18:00 фіксується різко виражений пік, який досягає значення понад 18 одиниць, що значно перевищує фоновий рівень і вказує на наявність аномальної події. Подібні різкі сплески можуть бути пов’язані з активацією прихованого функціоналу, встановленням зовнішнього з’єднання або короткочасною передачею специфічних пакунків. Додаткові, хоча й менш виражені піки спостерігаються ближче до 20:00. Загальний вигляд графіка свідчить про те, що окремі інтервали часу мають суттєві відхилення від типового профілю, що робить їх потенційно важливими для подальшого аналізу на предмет можливих ознак прихованої C2-активності або тестових шкідливих операцій. Методологія дослідження була сформована з урахуванням специфіки загроз, пов’язаних із використанням альтернативних або модифікованих прошивок маршрутизаторів, у яких можуть бути інтегровані приховані C2-механізми, елементи архітектури block-safe та модулі тривалої прихованої активності. Комплексний підхід об’єднував три взаємодоповнюючі напрями: динамічний аналіз трафіку, порівняння

контрольних сум прошивок та часткову зворотну інженерію бінарних компонентів. Перший напрям передбачав розгортання ізольованого тестового стенду з маршрутизаторами, на які встановлювались оригінальні, відкриті та кастомні прошивки. У межах цього стенду здійснювався збір тривалих сесій трафіку під типовим навантаженням (веб-навігація, стрімінг, VPN, оновлення), а також профілювання об'єму й періодичності з'єднань. Такий підхід дозволив визначати аномальні патерни, характерні для модифікованих образів, зокрема повторювані DNS-/HTTP-запити, нестандартні порти та звернення до доменів другого/третього рівнів, які не є типовими для стандартної прошивки [11-14]. Виявлені періодичні вихідні з'єднання стали ключовою ознакою потенційних прихованих C2-каналів, описаних у сучасних дослідженнях IoT-ботнетів [12], [18]. Другий напрям стосувався порівняння контрольних сум прошивок. Було досліджено оригінальні образи виробників, відкриті альтернативи та кастомні рішення невідомого походження. Обчислення MD5 і SHA-256 дозволило ідентифікувати явно модифіковані образи, зіставити їх зі сторонніми репозиторіями та визначити випадки підміни легітимних компонентів на шкідливі модулі.

Додатковий аналіз файлової структури (SquashFS, JFFS2) дав змогу виявити нові бінарні файли або змінені системні компоненти, що відповідає висновкам робіт, присвячених аналізу шкідливих прошивок у sandbox-середовищах [14], [17]. Третій напрям передбачав базову зворотну інженерію підозрілих модулів шляхом дослідження таблиць імпорту, пошуку рядкових констант, аналізу викликів мережевих бібліотек та відновлення логіки роботи елементарних автоматів. Такий аналіз дозволив ідентифікувати вбудовані ключі шифрування, шаблони доменних імен для генерації DNS-запитів та алгоритми побудови пакетів-тригерів. Аналогічні механізми описуються в дослідженнях, що розглядають еволюцію IoT-ботнетів та адаптивну поведінку прихованих модулів [15], [18–20]. Застосована методологія забезпечила багаторівневий підхід до виявлення аномалій у прошивках маршрутизаторів. Поєднання аналізу мережевої активності, перевірки цілісності та часткової реконструкції логіки роботи модулів дозволило виявити різні типи прихованої активності — від аномальних зовнішніх сесій до інтегрованих шкідливих компонентів усередині прошивки. Це підтверджує ефективність комплексних інструментів для аналізу сучасних модифікованих прошивок, які використовують шифрування, нестандартну періодичність і архітектуру block-safe для уникнення виявлення [18–20]. Спираючись на результати застосованої методології, у наступному розділі наведено виявлені ознаки прихованих C2-модулів, характерні патерни трафіку та реконструйовані механізми активації, що свідчать про можливу ботнет-орієнтовану функціональність у досліджуваних альтернативних прошивках.

Результати експериментального аналізу

Експериментальний аналіз альтернативних та модифікованих прошивок маршрутизаторів був виконаний як у контрольованому стендовому середовищі, так і на основі реальних даних SOzC, що включали NetFlow-записи, журнали NAT/firewall та фрагменти pcap-трейсів. Сукупність цих джерел дозволила виділити характерні ознаки прихованих C2-механізмів, визначити алгоритми їх активації, а також реконструювати роль заражених маршрутизаторів у ботнет-мережах. Результати експериментів підтверджують наявність змінених модулів у прошивках, їх здатність до встановлення прихованих каналів взаємодії та участі в DDoS-атаках.

Типові ознаки прихованих C2-механізмів

Порівняння оригінальних, відкритих та кастомних прошивок показало, що альтернативні модифіковані образи часто містять приховані бінарні модулі, які не використовуються штатною системою та не документуються виробниками. У більшості випадків ці модулі запускаються через cron із періодичністю 5-15 хвилин і генерують короткі, але регулярні шифровані TCP/UDP-з'єднання до доменів другого й третього рівнів, які не пов'язані з офіційними сервісами. У половині зразків спостерігалося використання нестандартних портів

у діапазоні 10000–20000, що не відображалися в інтерфейсі керування, але були зафіксовані в таблицях NAT.

Таблиця 1

Ознаки прихованих C2-механізмів у модифікованих прошивках

Ознака	**Експериментальний результат**	**Інтерпретація**
Додаткові приховані модулі	Виявлені у 78% аналізованих прошивок	Інтеграція стороннього функціоналу
Періодичні cron-виклики	Інтервали 5--15 хв	Канал підтримки зв’язку з C2
Сесії до доменів .blocksafe.net, .sync-mesh.io	Зареєстровані у 62% пристроїв	Можливі вузли C2-інфраструктури
Використання портів 10000--20000	У 51% випадків	Уникнення сигнатурного виявлення
Обсяг трафіку у «сплячому» режимі	0.3--0.6% від середньодобового	Маскування під системні запити

Узагальнення: комбінація періодичних фонових з’єднань, прихованих модулів і нестандартних портів є системним індикатором прихованого C2-механізму.

Пакет-тригер як механізм активації шкідливого функціоналу

Аналіз pcap-трейсів показав, що модулі прихованої активності чітко розмежовують пасивну фазу (фонова низькооб’ємна активність) та активну фазу (прийом команд C2). Перехід між фазами відбувається у відповідь на надходження пакет-тригера, структура якого залежить від типу шкідливого модуля.

Було виявлено три основні типи тригерів:

1. HTTP-запити з нетиповими параметрами;
2. DNS-відповіді зі зміненими полями;
3. UDP-пакети на нестандартні порти з фіксованою сигнатурою.

Після активації модуль негайно встановлює довготривалі зашифровані з’єднання з C2-інфраструктурою, отримує параметри атаки або маршрутизації і переходить у режим виконання команд. На графіках інтегральної аномальної оцінки такі події проявляються у вигляді різких піків у 15–40 разів вищих за фоновий рівень.

Таблиця 2

Виявлені шаблони активації пакет-тригерів

Тип тригера	**Параметри**	**Реакція модуля**	**Наслідки**
HTTP-тригер	GET /update?cfg=hex…, змінений User-Agent	Активація за 1--2 с	Встановлення C2-каналу
DNS-тригер	Аномальні значення TTL, Additional section	Порівняння із внутрішнім шаблоном	Перехід FSM у стан «active»
UDP-тригер	Порт 14567, сигнатура перших 8 байтів	Повторне завантаження модуля	Підготовка до DDoS або тунелю

Узагальнення: наявність пакет-тригера з фіксованою структурою є прямим доказом наявності загорнутого FSM-механізму активації, характерного для block-safe прошивок.

Інтеграція до ботнет-мереж і участь у DDoS-атаках

На основі логів C2-взаємодії та реальних NetFlow-трас SOzC було змодельовано типовий життєвий цикл маршрутизатора в ботнеті. Після активації модуль надсилає на C2-вузол дані про пристрій: унікальний ID, версію прошивки, обсяг пам'яті та доступні інтерфейси. Відповідно до цих параметрів інфраструктура розміщує пристрій у певному пулі (вузли з високою пропускною здатністю, вузли із доступом у внутрішні мережі, вузли для UDP flood тощо). Під час DDoS-кампаній вихідний трафік маршрутизаторів із модифікованими прошивками різко зростає: інтенсивність потоків збільшується у 8–25 разів, з'являються однотипні короткі сесії до обмеженої множини IP-адрес, а інколи виконується приховане перенаправлення NAT-правил.

Таблиця 3

Поведінка маршрутизаторів під час C2-команд і DDoS-активації

Етап	Зафіксована поведінка	Параметри атаки	Наслідки
Реєстрація в ботнеті	Передача ID, конфігурації, ресурсів	—	Додавання у відповідний пул
Отримання команд	Постійні TCP-сеанси з C2	SYN/UDP flood, HTTP flood	Різкий ріст трафіку
Активна фаза	Збільшення вихідних потоків у 8--25 разів	30--120 с активності	Участь у DDoS
Повернення у пасивний режим	Очищення тимчасових даних	—	Маскування слідів атаки

Узагальнення: модифіковані прошивки не лише забезпечують приховане підключення до C2-інфраструктури, а й реалізують короткотривалі, але високоефективні DDoS-атаки, після яких пристрій повертається у «чистий» режим.

Загальний висновок до розділу «Результати»

Результати дослідження однозначно підтверджують, що альтернативні модифіковані прошивки маршрутизаторів можуть містити вбудовані приховані модулі, механізми фонової періодичної взаємодії з C2-вузлами, алгоритми активації через пакет-тригери та логіку участі в DDoS-атаках. Реальні дані SOzC узгоджуються зі стендовими експериментами та демонструють такі самі поведінкові ознаки, що дозволяє говорити про промислове використання таких прошивок у ботнет-мережах.

Дискусія

Отримані результати показують, що альтернативні прошивки маршрутизаторів формують багаторівневі приховані C2-модулі, які важко виявити стандартними засобами моніторингу. DNS-трафік демонструє використання DGA-доменів та циклічних коротких запитів, що маскують сигнальні пакети для підтримання каналу взаємодії з C2. Кореляція між такими сигналами та піками вихідного трафіку свідчить, що запуск DDoS-функцій здійснюється через спеціальні структурні пакет-тригери, які не залежать від обсягу трафіку, а лише від конфігурації заголовків. Після активації модулі різко збільшують інтенсивність SYN- та UDP-флудів, а потім очищують тимчасові дані, що ускладнює форензик і залишає лише фрагментарні сліди в NetFlow або firewall-логах. Це підтверджує перехід IoT-ботнетів до моделі максимальної латентності, невидимості та мінімальної зміни конфігурації пристрою. Аналіз наукових робіт свідчить, що ефективне виявлення можливе лише за допомогою гібридних ML-моделей, які одночасно враховують мережеві ознаки, структуру прошивки та

поведінковий контекст пристрою. Окремо взяті мережевий чи статичний аналізи не дають повної картини. Лише інтегрований підхід дозволяє відстежити повний життєвий цикл шкідливого модуля – від латентної реєстрації до DDoS-атаки й подальшого маскування.

Таким чином, маршрутизаторні ботнети – це високорозвинені адаптивні системи з DGA-механізмами, структурними тригерами та стратегією очищення даних, що робить традиційні засоби виявлення малоефективними. Необхідні спеціалізовані firmware-aware IDS, які поєднують аналіз логіки прошивки й поведінкові моделі трафіку для реального захисту критичних мережевих сегментів.

Висновки

Проведене дослідження підтвердило, що альтернативні та модифіковані прошивки маршрутизаторів становлять окремий, високоризиковий клас загроз, який відіграє ключову роль у формуванні сучасних ботнет-мереж та реалізації DDoS-атак. На відміну від традиційних шкідливих програм, приховані модулі вбудовуються безпосередньо у файлову структуру прошивки, що забезпечує поступове, непомітне та стійке проникнення в мережеве середовище користувача. Отримані результати експериментального аналізу, виконаного на стендових даних та NetFlow-трейсах реальної інфраструктури, показали, що модифіковані прошивки містять типові ознаки прихованих C2-каналів: фонові шифровані TCP/UDP-виклики, циклічні cron-запуски невідомих бінарних модулів, використання нестандартних портів, а також звернення до нетипових доменних зон. Ці характеристики формують складну систему прихованої взаємодії, що дозволяє підтримувати довготривалу латентність та уникати виявлення стандартними IDS/IPS-системами. Ключовим відкриттям стало виявлення механізму пакет-тригера, який слугує точкою активації переходу прошивки зі «сплячого» стану до активної фази. Тригери у вигляді специфічних HTTP-, DNS- або UDP-пакетів запускають шкідливий модуль, ініціюють встановлення C2-сеансу та активують DDoS-функціонал. Така поведінка узгоджується з концепцією block-safe архітектури та підтверджує використання FSM-подібних механізмів у прихованих прошивках. Аналіз життєвого циклу маршрутизаторів у складі ботнетів засвідчив, що компрометовані пристрої не лише отримують команди, а й автоматично розподіляються за ролями залежно від пропускної здатності, конфігурації та мережевих прав доступу. Під час DDoS-кампаній інтенсивність вихідного трафіку таких пристроїв збільшується у 8–25 разів, що дозволяє організовувати масштабні атаки навіть за наявності невеликої кількості активних вузлів. Після завершення атаки прошивка повертається до «чистого» режиму, очищаючи тимчасові дані та мінімізуючи сліди інциденту. Сформований у роботі алгоритм виявлення компрометації маршрутизаторів довів свою ефективність і може бути використаний операторами мереж та службами кібербезпеки. Інтеграція NetFlow-аналізу, перевірки контрольних сум прошивок, кореляції подій у SIEM та динамічного аналізу доменної активності дозволяє точно ідентифікувати ознаки прихованої взаємодії з C2-інфраструктурою. Практична значущість отриманих результатів полягає у формуванні рекомендацій з управління ризиками, зокрема: централізованого менеджменту оновлень, сертифікації допустимих образів прошивок, обов'язкової атестації мережевого обладнання у критичній інфраструктурі та підвищення рівня обізнаності користувачів і персоналу. Це дозволяє значно знизити ймовірність компрометації та посилити стійкість мереж до ботнет-активності. У цілому, проведене дослідження підкреслює важливість переходу від суто мережевого або статичного аналізу до багаторівневого firmware-aware підходу, що враховує як структуру прошивки, так і поведінку трафіку. Саме такий інтегрований підхід здатний забезпечити раннє виявлення шкідливих модулів, нейтралізацію загроз block-safe архітектури та мінімізацію ризиків для приватних, корпоративних та критичних інфраструктур.

Перелік посилань

1. Tariq U., Ahanger T. A. Employing SAE-GRU deep learning for scalable botnet detection in smart city infrastructure. PeerJ Computer Science. 2025. Vol. 11. e2869. DOI: https://doi.org/10.7717/peerj-cs.2869.

2. Asadi M., Jamali M. A. J., Heidari A., Navimipour N. J. Botnets Unveiled: A Comprehensive Survey on Evolving Threats and Defense Strategies. Transactions on Emerging Telecommunications Technologies. 2024. Vol. 35. e5056. DOI: https://doi.org/10.1002/ett.5056.

3. Sivanesh S., Mani G., Senthilkumar D., Serrano S. BotCB: Unmasking Botnets Through Intelligent Network Traffic Analysis. International Journal of Distributed Sensor Networks. 2025. Article ID 1550147725. DOI: https://doi.org/10.1155/dsn/2344785.

4. Khaliq Z., Khan D. A., Baba A. I., Ali S., Farooq S. U. Model-based framework for exploiting sensors of IoT devices using a botnet: a case study with Android. Cyber-Physical Systems. 2025. Vol. 11, No. 1. P. 1–46. DOI: https://doi.org/10.1080/23335777.2024.2350001.

5. Baruah S., Deka V., Das D., Barman U., Saikia M. J. Enhanced Peer-to-Peer Botnet Detection Using Differential Evolution for Optimized Feature Selection. Future Internet. 2025. Vol. 17, No. 6. P. 247. DOI: https://doi.org/10.3390/fi17060247.

6. Alexander R., Pradeep Mohan Kumar K. BOTSIAM-DRL: Botnet detection using a few shot active matching siamese network deep reinforcement learning in IoT networks. Cluster Computing. 2025. Vol. 28, No. 10. P. 665. DOI: https://doi.org/10.1007/s10586-025-05497-5.

7. Almousa O., Hamdallh B., Al-nu'man R. Enhancing IoT Security: A Comparative Analysis of Machine Learning and Deep Learning Techniques for Botnet Detection. Engineering, Technology & Applied Science Research. 2025; 15(4):24498-24505. DOI: https://doi.org/10.48084/etasr.11092.

8. Ullah S., Wu J., Lin Z., Kamal M. M., Mostafa H., Sheraz M., Chuah T. C. Comparative analysis of deep learning and traditional methods for IoT botnet detection using a multi-model framework across diverse datasets. Scientific Reports. 2025; 15:31072. DOI: https://doi.org/10.1038/s41598-025-16553-w.

9. Asiri M., Khemakhem M. A., Alhebshi R. M., Alsulami B. S., Eassa F. E. Decentralized Federated Learning for IoT Malware Detection at the Multi-Access Edge: A Two-Tier, Privacy-Preserving Design. Future Internet. 2025. Vol. 17, No. 10. Article 475. DOI: https://doi.org/10.3390/fi17100475.

10 Niu G., Zhang F., Guo M. Terminal Forensics in Mobile Botnet Command and Control Detection Using a Novel Complex Picture Fuzzy CODAS Algorithm. Symmetry. 2025. Vol. 17, No. 10, p. 1637. DOI: https://doi.org/10.3390/sym17101637.

11. Shu J., Lu J. Two-Stage Botnet Detection Method Based on Feature Selection for Industrial Internet of Things. IET Information Security. 2025. Vol. ??, No. ??, pp. ??-??. DOI: https://doi.org/10.1049/ise2/9984635.

12. Kayyidavazhiyil A. Combined Tri-Classifiers for IoT Botnet Detection with Tuned Training Weights. International Journal of Image and Graphics. 2023. Vol. 25, No. 02. DOI: https://doi.org/10.1142/S021946782550007X.

13. Antony V., Thangarasu N. Chaotic crow search enhanced CRNN: a next-gen approach for IoT botnet attack detection. Indonesian Journal of Electrical Engineering and Computer Science. 2025. Vol. 38, No. 3, pp. 1745-1754. DOI: https://doi.org/10.11591/ijeecs.v38.i3.pp1745-1754.

14. Mutar M. H., El Fawal A. H., Nasser A., Mansour A. Predicting the Impact of Distributed Denial of Service (DDoS) Attacks in Long-Term Evolution for Machine (LTE-M) Networks Using a Continuous-Time Markov Chain (CTMC) Model. Electronics. 2024. Vol. 13, No. 21, Article 4145. DOI: https://doi.org/10.3390/electronics13214145.

15. Mohamed Saied, Shawkat Guirguis, Magda Madbouly. Review of filtering based feature selection for Botnet detection in the Internet of Things. Artificial Intelligence Review. 2025; Vol. 58, Article 119. DOI: https://doi.org/10.1007/s10462-025-11113-0.

16. Jannatul Ferdous, Rafiqul Islam, Arash Mahboubi, Md Zahidul Islam. A Survey on ML Techniques for Multi-Platform Malware Detection: Securing PC, Mobile Devices, IoT, and Cloud Environments. Sensors. 2025; Vol. 25, No. 4:1153. DOI: https://doi.org/10.3390/s25041153.

17. Gelgi M., Guan Y., Arunachala S., Samba Siva Rao M., Dragoni N. Systematic Literature Review of IoT Botnet DDoS Attacks and Evaluation of Detection Techniques. Sensors. 2024. Vol. 24, No. 11, Article 3571. DOI: https://doi.org/10.3390/s24113571.

18. Alqahtani M., Mathkour H., Ben Ismail M. M. IoT Botnet Attack Detection Based on Optimized Extreme Gradient Boosting and Feature Selection. Sensors. 2020. Vol. 20, No. 21. P. 6336. DOI: https://doi.org/10.3390/s20216336.

19. Almuqren L., Alqahtani H., Aljameel S. S., Salama A. S., Yaseen I., Alneil A. A. Hybrid Metaheuristics With Machine Learning Based Botnet Detection in Cloud Assisted Internet of Things Environment. DOI: https://doi.org/10.1007/s44196-022-00138-y.

20. Tariq U., Ahanger T. A. Employing SAE-GRU deep learning for scalable botnet detection in smart city infrastructure. PeerJ Computer Science. 2024. Article 2869. DOI: https://doi.org/10.7717/peerj-cs.2869.

21. AL-Azzawi W. S. A.-B., Hilou H. W., Warush N. H., Meslmani H., El-Douh A. A., Abdelhafeez A. A. Neutrosophic Set and Machine Learning Model for Identifying Botnet Attacks on IoT Effectively. Zenodo. 2024. DOI: https://doi.org/10.5281/zenodo.15717644.

Надійшла 16.11.2025