Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Інститут електроенергетики
Факультет інформаційних технологій
Кафедра безпеки інформації та телекомунікацій

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеню магістра

студента ***Горошко Євгена Олександровича***

академічної групи ***125м-17-1***

спеціальності ***125 Кібербезпека***

спеціалізації[1] ***Кібербезпека***

за освітньо-професійною програмою ***Кібербезпека***

на тему ***Способи протидії бот-мережам в системах інтернет речей***

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	д.т.н., проф. Корнієнко В.І.			
розділів:				
спеціальний	ст. викл. Кручінін О.В.			
економічний	к.е.н., доц. Пілова Д.П.			

Рецензент				

Нормоконтролер	ст. викл. Мєшков В.І.			

Дніпро
2018

ЗАТВЕРДЖЕНО:
завідувач кафедри
безпеки інформації та телекомунікацій
_____ д.т.н., проф. Корнієнко В.І.

«_____»_____20___року

ЗАВДАННЯ
на кваліфікаційну роботу ступеня магістра

студенту ***Горошко Є.О.*** **академічної групи** ***125м-17-1***

(прізвище та ініціали) (шифр)

спеціальності ***125 Кібербезпека***

спеціалізації[1] ***Кібербезпека***

за освітньо-професійною програмою ***Кібербезпека***

на тему ***Способи протидії бот-мережам в системах Інтернет речей***

1 ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ

Наказ ректора НТУ «Дніпровська політехніка» від 29.11.2018 р. № 2025-Л

2 МЕТА ТА ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБІТ

Об'єкт досліджень ***ботнет у системах інтернет речей***

Предмет досліджень ***методології протидії ботнет у системах інтернет речей.***

Мета ***розробка методології протидії ботнет у системах***

інтернет речей

Вихідні дані для проведення роботи *матеріали науково – дослідної та преддипломної практик*

3 ОЧІКУВАНІ РЕЗУЛЬТАТИ

Наукова новизна *запропонований метод аналізу мережевого трафіка пристроїв IoT, що забезпечує можливість виявлення аномалій та вторгнень у систему інтернету речей*

Практична цінність *результат роботи може слугувати основою для розробки системи виявлення вторгнень у систему IoT, що забезпечує захист системи від ботнет*

4 ВИМОГИ ДО РЕЗУЛЬТАТІВ ВИКОНАННЯ РОБОТИ

Відповідність методичним рекомендаціям до підготовки та захисту дипломної роботи та вимогам нормативним документів з технічного захисту інформації

5 ЕТАПИ ВИКОНАННЯ РОБІТ

Найменування етапів робіт	**Строки виконання робіт** (початок-кінець)
Огляд джерел за темою та напрям досліджень	03.09.18-06.10.18
Методи досліджень	07.10.18-31.10.18
Результати досліджень	01.11.18-24.11.18

Виконання економічного розділу	25.11.18-04.12.18
Оформлення пояснювальної записки	05.12.18-10.12.18

6 РЕАЛІЗАЦІЯ РЕЗУЛЬТАТІВ ТА ЕФЕКТИВНІСТЬ

Економічний ефект ***Запобігання фінансових витрат при створенні системи виявлення вторгнень для IoT на визначення необхідних параметрів мережевого трафіка***

Соціальний ефект ***Захищеність пристроїв IoT , що для користувачів є, безумовно, дуже важливим аспектом.***

7 ДОДАТКОВІ ВИМОГИ

Завдання видано _____ _____
(підпис керівника) (прізвище, ініціали)

Дата видачі: 03.09.18р.

Дата подання до екзаменаційної комісії: 14.12.18р.

Прийнято до виконання _____ _____
(підпис студента) (прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: ___ с., __рис., __ табл., __ додатків, __ джерел.

Об’єкт дослідження: ботнет у системах інтернет речей.

Предмет дослідження: способи протидії ботнет у системах інтернет речей.

Мета дипломної роботи: вдосконалення методів протидії ботнет у системах інтернет речей

У першому розділі розглянута технологія IoT, її захищеність, класифіковані загрози, а також розглянуте зловмисне програмне забезпечення – ботнет.

У другому розділі роботи розглянуті алгоритми роботи ботнет, класичні методи протидії ботнет, досліджені особливості систем IoT, проаналізовані параметри трафіку пристроїв IoT. Визначені найбільш інформативні параметри трафіку для точних результатів аналізу.

В економічній частині проведений розрахунок капітальних витрат на розробку алгоритму аналіза мережевого трафіка інтернет речей.

Наукова новизна полягає в запропонованому алгоритмі збору та аналізу мережевого трафіка пристроїв IoT, що забезпечує можливість виявлення аномалій та вторгнень у систему інтернету речей.

БОТНЕТ, ІНТЕРНЕТ РЕЧЕЙ, МЕТОДИ ПРОТИДІЇ БОТНЕТ, АНАЛІЗ МЕРЕЖЕВОГО ТРАФІКА.

РЕФЕРАТ

Пояснительная записка: ___стр., __ рис., __табл., __ приложений, __ источников.

Объект исследования: ботнет в системах интернет вещей.

Предмет исследования: способы противодействия ботнет в системах интернет вещей.

Цель дипломной работы: усовершенствование методов противодействия ботнет в системах интернет вещей.

В 1 разделе рассмотрена технология IoT, ее защищенность, классифицированны угрозы, а также рассмотрено вредоносное ПО - ботнет.

Во 2 разделе работы рассмотрены алгоритмы работы ботнет, классические методы противодействия ботнет, исследованы особенности систем IoT, проанализированы параметры трафика устройств IoT. Определены наиболее информативные параметры трафика для точных результатов анализа.

В экономической части произведен расчет капитальных затрат на разработку алгоритма анализа сетевого трафика интернет вещей.

Научная новизна заключается в предложенном алгоритме сбора и анализа сетевого трафика устройств IoT, что обеспечивает возможность выявления аномалий и вторжений в систему интернет вещей.

БОТНЕТ, ИНТЕРНЕТ ВЕЩЕЙ, МЕТОДЫ ПРОТИВОДЕЙСТВИЯ БОТНЕТ, АНАЛИЗ СЕТЕВОГО ТРАФИКА.

ABSTRACT

Explanatory note: ___ p., ___ fig., ___ tab., ___ additions, ___ sources.

The object of research: botnet in Internet of Things systems.

The purpose of research: methods to counter botnets in Internet of Things systems.

The aim of the thesis: enhancing methods to counter botnets in Internet of Things systems.

The first section discusses IoT technology, its security, threats were classified, also the malware botnet was considered.

In the second section of the paper there were the algorithms of the botnet, the classic methods of counteracting botnet considered, the features of the IoT systems were analyzed. Traffic features of the IoT devices analyzed. The most informative traffic parameters are defined for accurate analysis results.

In the economic part, the capital expenditures, for the development of the network traffic of IoT devices analysis algorithm were calculated.

The scientific novelty consists in proposing an algorithm for collecting and analyzing network traffic of IoT devices, which provides the possibility to detect anomalies and invasions into the Internet of things systems.

BOTNET, INTERNET THINGS, BOTNET COUNTER METHODS, NETWORK TRAFFIC ANALYSIS.

СПИСОК УМОВНИХ СКОРОЧЕНЬ

IoT – Інтернéт речéй;

RFID – радіочастотна ідентифікація;

TCP – Transmission Control Protocol;

ПЗ – програмне забезпечення;

DVR – Digital Video Recorder;

RCE – Remote Command Execution;

SSL – Secure Sockets Layer;

OTA – On the air;

SSH – Secure Shell

JTAG – Joint Test Action Group;

SWD – Serial Wire Debug;

SPI – Serial Peripheral Interface;

UART – Universal Asynchronous Receiver-Transmitter;

CLI – Command Line Interface;

UDP – User Datagram Protocol;

DoS – Denial of Service;

IRC – Internet Relay Chat;

P2P – peer-to-peer;

Telnet – Teletype network;

C&C – Command and Control;

FTP – File transfer protocol;

DHCP – Dynamic Host Configuration Protocol;

ЕОМ – Електронно-обчислювальна машина;

AV – anti-virus;

DNS – Domain Name System

ЗМІСТ

ВСТУП

Інтернет речей (англ. Internet of Things, IoT) — концепція мережі, що складається із взаємозв'язаних фізичних пристроїв, які мають вбудовані датчики, а також програмне забезпечення, що дозволяє здійснювати передачу і обмін даними між фізичним світом і комп'ютерними системами, за допомогою використання стандартних протоколів зв'язку. Окрім датчиків, мережа може мати виконавчі пристрої, вбудовані у фізичні об'єкти і пов'язані між собою через дротові чи бездротові мережі. Ці взаємопов'язані пристрої мають можливість зчитувати та обробляти дані для автоматизації процесів, дозволяючи виключити необхідність участі людини.

Кількість пристроїв Internet of Things (IoT), збільшується, але, як показують дослідження, багато з цих пристроїв є принципово небезпечними, наражаючи інтернет та всіх його користувачів на різноманітні атаки. Найпоширенішим методом використання пристроїв IoT є ботнети. Наприклад, такі ботнети як Mirai, використовують незахищені пристрої IoT, щоб здійснити атаки DDoS на критичну інтернет-інфраструктуру.

Таке становище мотивує до розробки нових методів автоматизації реагування на атаки IoT та протидії ботнет. У цій роботі виконаний аналіз, який підтверджує, що використання специфічних для IoT мережевих моделей поведінки (наприклад, обмежена кількість кінцевих точок або регулярні часові проміжки між передачею пакетів даних) може допомогти точно визначити параметри для збору та аналізу мережевого трафіка пристроїв IoT. Визначення таких параметрів є ключовим фактором в подальшому дослідженні та створенні системи виявлення вторгнень до систем IoT.

Така система має виключити можливість легкої компрометації та використання IoT пристроїв у ботнетах.

РОЗДІЛ 1. ЗАХИЩЕНІСТЬ IOT ТА АНАЛІЗ БОТНЕТ

1.1 Визначення та огляд технології IoT

Термін «Інтернет речей» (IoT) вперше був введений Кевіном Ештоном щоб описати систему, в якій фізичні об'єкти могли бути пов'язані з датчиками і мережею Інтернет. Ештон ввів цей термін, щоб проілюструвати можливості радіочастотної ідентифікації (*RFID*), яка використовується в корпоративних системах поставок, щоб порахувати і відстежити товари без потреби в людському втручанні. Сьогыодні, інтернет речей став популярним терміном для опису сценаріїв, у яких інтернет з'єднання і обчислювальна здатність поширюються на безліч об'єктів, пристроїв, датчиків і повсякденних об’єктів.

Основною концепцією IoT є можливість підключення всіляких об'єктів (речей), які людина може використовувати в повсякденному житті, наприклад, холодильник, кондиціонер, автомобіль, велосипед і навіть кросівки. Всі ці об'єкти (речі) повинні бути оснащені вбудованими датчиками або сенсорами, які мають можливість обробляти інформацію, що надходить з навколишнього середовища, обмінюватися нею і виконувати різні дії в залежності від отриманої інформації. Прикладом впровадження такої концепції є система «розумний будинок» або «розумна ферма». Ця система аналізує дані навколишнього середовища і в залежності від показників регулює температуру в приміщенні. У зимовий період регулюються інтенсивність опалення, а в разі спекотної погоди будинок має механізми відкривання і закривання вікон, завдяки чому провітрюється будинок, і все це відбувається без втручання людини.

Для об'єднання повсякденних речей у мережу потрібні декілька технологій:

- Для ідентифікації кожного об'єкту потрібна проста та компактна технологія. Тільки при наявності системи унікальної ідентифікації можна збирати та накопичувати інформацію про певний предмет. Такий функціонал можна забезпечити за допомогою мікросхем *RFID* (*Radio-Frequency IDentification*).

Вони здатні без власного джерела струму передавати інформацію приладам зчитування. Кожна мікросхема має індивідуальний номер. Як альтернатива до даної технології для ідентифікації об'єктів можуть використовуватись QR-коди. Для визначення точного місця знаходження речі підійде технологія *GPS*, яка ефективно використовується вже сьогодні у смартфонах та навігаторах.

- Для відслідковування змін у стані елементу чи оточуючого середовища об'єкти повинні оснащуватися сенсорами.

- Для обробки та накопичення даних з сенсорів повинен використовуватися вбудований обчислювальний пристрій маленьких розмірів (наприклад *Raspberry Pi, Intel Edison*)

- Для обміну інформацією між пристроями можуть бути використані технології бездротових мереж (*Wi-Fi, Bluetooth, ZigBee, 6LoWPAN*).

Інтеграція з Інтернетом має на увазі, що пристрої будуть використовувати IP-адресу як унікальний ідентифікатор. Проте, через обмежені адресні простори в *IPv4* (що дозволяє використовувати 4,3 мільярда унікальних адрес), об'єктам IP доведеться використовувати *IPv6*, який забезпечує унікальними адресами мережевого рівня не менше 300 млн пристроїв на одного жителя Землі. Об'єктами в IP будуть не тільки пристрої із сенсорними можливостями, але також пристрої, які виконують дії (наприклад, лампочки або замки, якими керують через Інтернет). Значною мірою, майбутнє інтернету речей не буде можливим без підтримки *IPv6*, отже, глобальне впровадження *IPv6* у найближчі роки буде мати вирішальне значення для успішного розвитку IP в майбутньому.

Для бездротової передачі даних особливо важливу роль в побудові інтернету речей відіграють такі характеристики, як ефективність, відмовостійкість, адаптивність, можливість самоорганізації. Основне зацікавлення в цьому сенсі представляє стандарт *IEEE 802.15.4*, що управляє доступом для організації енергоефективних персональних мереж, і є основою для таких протоколів, як *ZigBee, WiFi, Bluetoot, 6LoWPAN*.

Серед провідних технологій важливу роль у розповсюдженні інтернету речей відіграють рішення *PLC* — технології побудови мереж передачі даних по лініях електропередач, оскільки у багатьох додатках присутній доступ до електромереж (наприклад, торгові автомати, банкомати, інтелектуальні лічильники, контролери освітлення спочатку підключені до мережі електропостачання). *6LoWPAN*, який реалізує шар *IPv6* як над *IEEE 802.15.4*, так і над *PLC*, будучи відкритим протоколом, стандартизованих *IETF*, відзначається як особливо важливий для розвитку інтернету речей.

Вже зараз інтернету речей приділяється увага на найвищому рівні, зокрема починаючи з 2009 року у Брюселі при підтримці Єврокомісії проходять конференції Annual Internet of Things, на який виступають з доповідями єврокомісари, науковці та керівники провідних IT-компаній. За прогнозами аналітиків у найближчі роки очікується справжній бум інтернету речей. Так, за прогнозами Gartner, до 2020 року кількість підключених до всесвітньої мережі пристроїв становитиме 26 мільярдів, а дохід від продажу устаткування, програмного забезпечення та послуг становитиме 1,9 трлн дол. Деякі інші аналітичні агентства висловлюють ще більш оптимістичні прогнози. Найбільші світові IT компанії вже почали перегони за лідерство на цьому ринку. Так корпорація Intel у 2014 році після випуску «SoC Edison» оголосила конкурс «*Make it Wearable*» з призовим фондом $1,3 млн на найкраще застосування своєї системи для концепції IoT та створила власний підрозділ «Internet of Things Solutions Group» для розвитку цього напрямку. Компанія «Google» на початку 2014 року за 3,2 млрд доларів купила невелику фірму «Nest Labs», яка займається випуском інтелектуальних термостатів. Спеціалісти цієї компанії займались впровадженням на американському ринку технологій IoT. Виробники побутової техніки також працюють у цьому напрямку. Так на виставці CES 2014 у Лас-Вегасі була представлена велика кількість побутової техніки (холодильники, телевізори, пральні машини) з можливістю підключення до інтернет.

Лідерами у розробці та впровадженні інтернету речей є країни, в який розвинена індустрія виробництва мікропроцесорів та вбудованих комп'ютерів —

це США, Китай, Південна Корея. Також значний прогрес у цій галузі демонструють європейські країни та Японія.

Інтернет речей може викликати величезні зміни у повсякденному житті, надавши звичайним користувачам абсолютно новий рівень комфорту. Але якщо елементи такої системи не будуть належним чином захищені від несанкціонованого втручання, за допомогою надійного криптографічного алгоритму, замість користі вони принесуть шкоду, надавши кіберзлочинцям лазівку для підриву інформаційної безпеки. Оскільки речі із вбудованими комп'ютерами зберігають дуже багато інформації про свого власника, зокрема можуть знати його точне місцезнаходження, доступ до такої інформації може допомогти зловмисникам вчинити злочин. Відсутність на даний час стандартів для захисту таких автономних мереж дещо сповільнює впровадження інтернету речей у повсякденне життя.

Тож, у швидко-зростаючій галузі технологій, яка застосовується у всьому світі є численна кількість вразливостей. Виникють питання: які найбільш розповсюджені та критичні?

1.2 Аналіз захищеності технології IoT

Щоб зрозуміти нинішній стан безпеки досліджуваної технології треба проаналізувати та відповісти на три основні питання: звідки може надходити загроза – джерело загроз та його особливості; які на даний момент існують вразливості, та які найбільш поширені для використання; яким чином використовуються описані вразливості – атаки, які реалізують вразливості, та їх поширення.

1.2.1 Класифікація вразливостей

Компоненти розумного будинку та навіть охоронні системи у значній кількості мають масові та грубі порушення принципів розробки:

- використання незмінних (hardcoded) і прихованих сервісних облікових даних;
- застосування однакових або легко передбачуваних ключів та ПІН-кодів;
- відсутність перевірки прав доступу при зверненні до відомої сторінки налаштувань (наприклад, /settings.asp в обхід /index.htm) або прямого виклику зображень і відео потоку IP-камери (на кшталт /axis-cgi/jpg/image.cgi);
- некоректна обробка отриманих даних, що викликає переповнення буфера. Як наслідок, можливе виконання довільного коду при отриманні зловмисно складеного TCP-пакета;
- примусове перемикання сервера на використання старих версій протоколів за запитом клієнтського пристрою;
- десятки інших типових помилок і навмисних послаблень безпеки заради зручності конфігурації нефахівцями (в тому числі - віддаленого і без належної авторизації).

Уразливості наступних типів найбільш не захищені та для зловмисника дають більше гарантій успіху:

- виявлені після того, як виробник припинив підтримку пристрою і випуск патчів;
- виявлені недавно (для яких ще немає виправлень, або більшість користувачів не встигли виправлення застосувати);
- архітектурні недоліки, погано виправляються патчами ПЗ і рідко усуваються до кінця - на кшталт уразливості Spectre, існуючої в декількох різновидах і яка є актуальною досі;
- впливають відразу на кілька моделей і навіть типів пристроїв (наприклад, через загальне компонента веб-інтерфейсу або уразливості самого протоколу комунікації).

Нижче приведені 3 уразливості, які використовували найчастіше за перший квартал 2018 року:

- *MVPower DVR Remote Code Execution* .

На пристроях MVPower DVR інсує вразливість RCE (виконання коду віддалено). Віддалений атакуючий може використовувати цю слабкість для виконання довільного коду на ураженому маршрутизаторі за допомогою спеціально-розробленого запиту.

- *D-Link DSL-2750B Remote Command Execution.*

Повідомляється про уразливість RCE (можливість виконання коду віддалено) у роутерах D-Link DSL-2750B. Успішна експлуатація може призвести до виконання довільного коду на уразливому пристрої.

- *OpenSSL tls_get_message_body Function init_msg Structure Use After Free (CVE-2016-6309).*

Повідомляється про вразливість use-after-free функції OpenSSL – tls_get_message_body. Віддалений, неавтентифікований зловмисник може використати цю вразливість, відправивши створене повідомлення на уразливий сервер. Успішна експлуатація дозволяє зловмиснику виконати довільний код у системі.

Оксфорд визначає Інтернет речей як: "Пропонована розробка Інтернету, в якому повсякденні об'єкти мають підключення до мережі, що дозволяє їм надсилати та отримувати дані".

Проект OWASP Internet of Things розроблений, щоб допомогти виробникам, розробникам та споживачам краще зрозуміти проблеми безпеки, пов'язані з Інтернетом речей, і дозволити користувачам в будь-якому контексті приймати більш безпечні рішення при побудові, впровадженні та оцінці технологій IoT. Одною зі складових проекту є виявлення та аналіз найбільш поширених вразливостей (Таблиця 1.1).

Таблиця 1.1 – Аналіз найбільш поширених вразливостей IoT

Вразливість	Ціль атаки	Результат
Підбір username	• Адміністративний інтерфейс • Веб інтерфейс пристрою • Інтерфейс хмарного сховища • Мобільний додаток	• Можливість збирати набір валідних username, взаємодіючи з механізмом автентифікації

Продовження таблиці 1.1

Вразливість	Ціль атаки	Результат
Слабкі паролі	• Адміністративний інтерфейс • Веб інтерфейс пристрою • Інтерфейс хмарного сховища • Мобільний додаток	• Можливість встановити паролі '1234' або '123456' наприклад. • Використання встановленого за замовчуванням паролю
Вимкнене блокування акаунту	• Адміністративний інтерфейс • Веб інтерфейс пристрою • Інтерфейс хмарного сховища • Мобільний додаток	• Можливість продовжувати висилати запити після 3-5 неправильних спроб вводу пароля
Відсутність шифрування передачі даних	• Мережеві служби пристрою	• Мережеві служби не зашифровані правильним шляхом для захисту від прослуховування зловмисниками
Двофакторна автентифікація	• Адміністративний інтерфейс • Інтерфейс хмарного сховища • Мобільний додаток	• Нестача двофакторної автентифікаціїї, такої як токен або відбиток пальця
Неякісно реалізоване шифрування	• Мережеві служби пристрою	• Шифрування реалізоване, хоча налаштоване не належним чином, або не було правильно оновлене, наприклад, використання SSL v2
Оновлення без шифрування	• Механізм оновлення	• Оновлення передаються мережею без використання TLS або шифрування самого файлу оновлення
Місцезнаходження прошивок оновлення	• Механізм оновлення	• Місце зберігання файлів оновлень - легкодоступна інформація, що може дозволити модифікувати та розповсюджувати таку прошивку всім користувачам
DOS	• Мережеві служби пристрою	• Сервіс може бути атакований таким чином, що заперечує обслуговування для цієї служби або всього пристрою

Видалення інформації з пам’яті	• Фізичний інтерфейс пристрою	• Можливість фізичного видалення носія інформації з пристрою

Продовження таблиці 1.1

Вразливість	Ціль атаки	Результат
Немає ручного режиму оновлення	• Механізм оновлення	• Немає можливості вручну перевірити оновлення пристрою
Відсутніть можливості оновлення	• Механізм оновлення	• Немає можливості оновити пристрій
Висвічування дати оновлення або версії прошивки	• Прошивка пристрою	• Не відображається встановлена прошивка і / або остання дата оновлення
Прошивка та зчитування даних з пам'яті	• JTAG / SWD інтерфейс • Перехват OTA оновлення • Завантаження з вебсторінки виробників • Розпаювання флешчіпу пам'яті та зчитування через адаптер	• Прошивка має велику кількість корисної інформації, як вихідний код, сведення про служби, які використовуються, встановлені паролі, ключі SSH і т.п.
Маніпуляція з алгоритмом виконавчого кода пристроя	• JTAG / SWD інтерфейс • Атаки з інших каналів	• За допомогою JTAG адаптера та GDB можна модифікувати виконувану прошивку пристрою та обходити майже всі програмні захисні обмеження • Атаки з побічного каналу також можуть модифікувати виконуваний потік коду або можуть зливати інформацію з пристрою
Отримання доступу до консолі керування	• Серійні інтерфейси (SPI / UART)	• Підключившись до послідовного інтерфейсу, можна отримати повний доступ до консолі пристрою • Зазвичай захисні обмеження включають спеціальні загрузники, які запобігають вхід нападником в аккаунт користувача, але це тех можна обійти
Не сертифіковані частини ПЗ	• Програмне забезпечення	• Застарілі версії busybox, openssl, ssh, web servers тощо.

1.2.2 Класифікація основних атак

Оксфордский проект OWASP проводить методичні системні дослідження безпеки IoT та їх складових. На основі досліджень були складені всі сфери (складові) IoT, на які може бути націлена атака, та засоби, які можуть бути використані (Таблиця 1.2).

Таблиця 1.2 – Основні цілі атаки та вразливості IoT

Ціль атаки	Вразливість
Контроль доступу до екосистеми	• Неявна довіра між компонентами • Запис безпеки • Система зняття з експлуатації • Загублені процедури доступу
Сховище пристрою	• Cleartext ім'я користувача • Cleartext паролі • Сторонні аккаунти • Ключі шифрування
Фізичні інтерфейси пристрою	• Вилучення прошивки • CLI користувача • CLI адміністратора • Привілейована ескалація • Скидання в небезпечний стан • Видалення носіїв інформації
Веб-інтерфейс пристрою	• SQL-ін’єкція • Перехресні сценарії • Перехресний запит на підробку • Перелік імен користувачів • Слабкі паролі • Блокування облікового запису • Відомі стандартні облікові дані

Прошивка пристрою	• Запрограмовані облікові дані • Розголошення конфіденційної інформації • Розголошення чутливої URL-адреси • Ключі шифрування • Відображення версії вбудованого програмного забезпечення та / або останню дату оновлення

Ціль атаки	Вразливість
Мережеві служби пристрою	• Розкриття інформації • CLI користувача • Адміністративний CLI • Ін'єкція • Відмова в обслуговуванні • Незашифровані сервіси • Погано реалізоване шифрування • Тестові / Розробницькі служби • Переповнення буфера • UPnP • Уразливі служби UDP • DoS
Адміністративний інтерфейс	• SQL ін'єкція • Перехресні сценарії • Перехресний запит на підробку • Перелік імен користувачів • Слабкі паролі • Блокування облікового запису • Відомі стандартні облікові дані • Параметри безпеки / шифрування • Параметри ведення журналу • Двофакторна автентифікація • Неможливість стерти налаштування пристрою
Хмарний веб-інтерфейс	• SQL ін’єкція • Перехресні сценарії • Перехресне запит на підробку • Перелік імен користувачів • Слабкі паролі • Блокування облікового запису • Відомі стандартні облікові дані • Транспортне шифрування • Незахищений механізм відновлення пароля • Двофакторна автентифікація

Продовження таблиці 1.2 – Основні цілі атаки та вразливості

Продовження таблиці 1.2 – Основні цілі атаки та вразливості

Ціль атаки	Вразливість
Сторонні API	• Незашифрований PII відправлений • Зашифрований PII відправлений • Інформація про пристрій вийшла • Місце витоку
Механізм оновлення	• Оновлення відправлено без шифрування • Оновлення не підписані • Відоме місцезнаходження оновлення • Оновлення перевірки • Шкідливе оновлення • Відсутній механізм оновлення • Немає ручного механізму оновлення
Мобільний додаток	• Неявно довіряють пристрою чи хмара • Перелік імен користувачів • Блокування облікового запису • Відомі стандартні облікові дані • Слабкі паролі • Незахищене зберігання даних • Транспортне шифрування • Незахищений механізм відновлення пароля • Двофакторна автентифікація
API провайдера	• Внутрішня довіра до хмарної або мобільної програми • Слабка автентифікація • Слабкі контролери доступу • Ін'єкційні напади
Комунікація в екосистемі	• Екосистемні команди • Виправлення • Натискання оновлень
Мережевий трафік	• LAN • LAN до Інтернету • Короткий діапазон • Не стандартизований

Також Kaspersky Lab опублікували результати дослідження загроз та атак на IoT. Вони стверджують, що інтерес до різних IoT-пристроїв з боку зловмисників продовжує зростати: за першу половину 2018 роки було отримано в три рази більше зразків шкідливого ПЗ, атакуючого «розумні» пристрої, ніж за весь 2017 рік. До слова, в 2017 їх було в 10 разів більше, ніж в 2016 році (Рисунок 1.1).

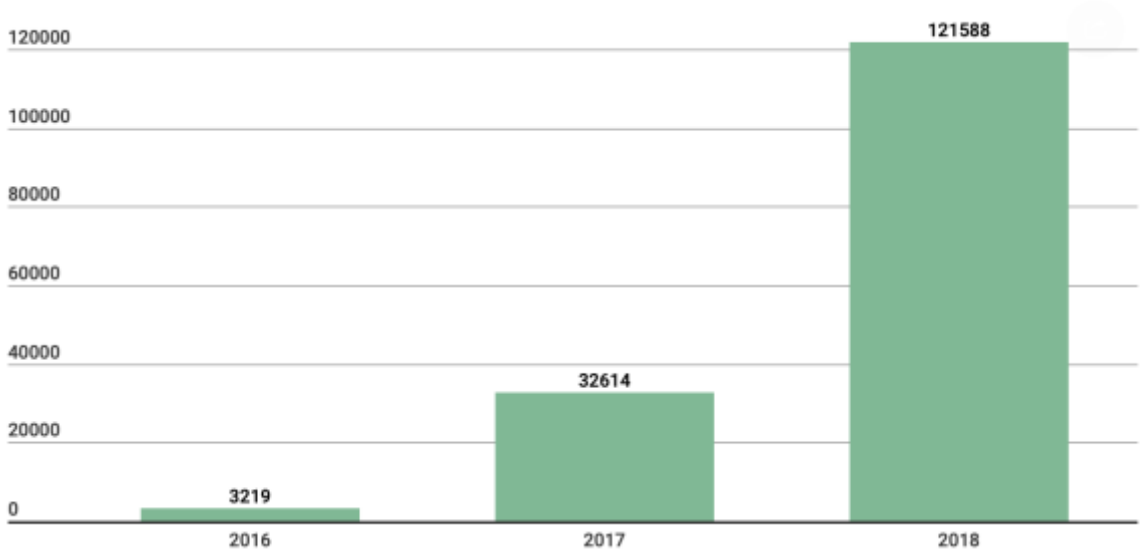


Рисунок 1.1 - кількість зразків шкідливого ПЗ для IoT-пристроїв в колекції «Лабораторії Касперського», 2016-2018 р.р.

Kaspersky Lab вирішили вивчити, які вектори атак використовуються зловмисниками для зараження «розумних» пристроїв, які шкідники завантажуються в систему в результаті успішної атаки, а також чим все це може обернутися для власника пристрою і просто жертв тільки-створеного ботнету.

Одним з найпопулярніших векторів атак і, відповідно, зараження пристроїв все ще залишається перебір пароля Telnet. У другому кварталі 2018 року таких атак на ханіпоти Kaspersky Lab було в 3 рази більше, ніж всіх інших разом узятих (Таблиця 1.3, 1.4).

Таблиця 1.3 – Відношення атакованих служб за даними Kaspersky Lab

Сервіс	% атак

Telnet	75,40 %
SSH	11,59 %
Інше	13,01 %

Оскільки частина власників «розумних» пристроїв змінює штатний пароль Telnet на більш складний, а багато гаджети і зовсім не підтримують цей протокол, зловмисники знаходяться в постійному пошуку нових шляхів зараження. Цьому сприяє і висока конкуренція між розробниками вірусів, яка призвела до зниження ефективності атак з перебором пароля: в разі успішного злому пристрою його пароль змінюється, а доступ до Telnet блокується.

Таблиця 1.4 - топ 10 шкідників, які завантажують на заражений IoT - пристрій в результаті успішного підбору пароля Telnet

#	Завантажений шкідник	% атак
1	Backdoor.Linux.Mirai.c	15,97 %
2	Trojan-Downloader.Linux.Hajime.a	5,89 %
3	Trojan-Downloader.Linux.NyaDrop.b	3,34 %
4	Backdoor.Linux.Mirai.b	2,72 %
5	Backdoor.Linux.Mirai.ba	1,94 %
6	Trojan-Downloader.Shell.Agent.p	0,38 %
7	Trojan-Downloader.Shell.Agent.as	0,27 %
8	Backdoor.Linux.Mirai.n	0,27 %
9	Backdoor.Linux.Gafgyt.ba	0,24 %
10	Backdoor.Linux.Gafgyt.af	0,20 %

Як приклад використання «альтернативної техніки» можна привести ботнет Reaper, в його активі станом на кінець 2017 року налічувалося близько 2 мільйонів IoT-пристроїв. Замість перебору паролів до Telnet цей ботнет експлуатував відомі уразливості в ПЗ:

- в прошивці роутера D-Link 850L;

- в IP-камерах GoAhead;
- в CCTV камерах MVPower;
- в Netgear ReadyNAS Surveillance;
- в Vacron NVR;
- в пристроях Netgear DGN;
- в роутерах Linksys E1500 / E2500;
- в роутерах D-Link DIR-600 і DIR 300 - HW rev B1;
- в пристроях AVTech.

Переваги такого способу поширення в порівнянні з перебором пароля:

- Зараження відбувається значно швидше;
- Закрити вразливість в ПЗ значно складніше, ніж змінити пароль або відключити / заблокувати сервіс.

Незважаючи на те, що цей спосіб складніший в реалізації, він припав до смаку багатьом розробникам вірусів і нові троянці, що використовують відомі уразливості в ПЗ «розумних» пристроїв, не змусили себе довго чекати.

Щоб визначити, які уразливості намагаються використовувати шкідники, було проаналізувано дані про спроби підключень до різних портів на ханіпотах (Таблиця 1.5).

Таблиця 1.5 – дані за другий квартал 2018 року

Служба	Порт	% атак	Вектор атак	Сімейство шкідливого ПЗ
Telnet	23, 2323	82 %	Брутфорс	Mirai, Gafgyt
SSH	22	12 %	Брутфорс	Mirai, Gafgyt
Samba	445	3 %	EternalBlue, EternalRed, CVE-2018-7445	—
tr-069	7547	1 %	RCE в реалізації протокола TR-069	Mirai, Hajime

HTTP	80	1 %	Спроби проексплуатувати вразливості у веб-сервері або підібрати пароль від панелі адміністрування	—
winbox (RouterOS)	8291	1 %	Використовується для ідентифікації RouterOS (MikroTik) і для атак через сервіс winbox	Hajime
Mikrotik http	8080	0 %	RCE в MikroTik RouterOS < 6.38.5 Chimay-Red	Hajime
MSSQL	1433	0 %	Виконання довільного коду для певних версій (2000, 2005, and 2008) або зміна пароля адміністратора, крадіжка даних	—
GoAhead httpd	81	0 %	RCE в IP-камерах GoAhead	Persirai, Gafgyt

Продовження таблиці 1.5

Служба	Порт	% атак	Вектор атак	Сімейство шкідливого ПЗ
Mikrotik http	8081	0 %	Chimay-Red	Hajime
Etherium JSON-RPC	8545	0 %	Обход авторизації (CVE-2017-12113)	—
RDP	3389	0 %	Брутфорс	—
XionMai uc-httpd	8000	0 %	Виконання довільного коду для питань комерційної торгівлі версій (2000, 2005, and 2008) або зміна пароля адміністратора, крадіжка Даних	Satori
MySQL	3306	0 %	Виконання довільного коду для певних версій (2000, 2005, and 2008) або зміна пароля адміністратора, крадіжка даних	—

Поки переважна більшість атак - це все ще перебір паролів Telnet і SSH. На третьому місці за популярністю знаходяться атаки на сервіс SMB, що надає віддалений доступ до файлів. Поки не було помічено IoT-шкідників, що атакують цей сервіс. Однак деякі його версії містять серйозні відомі уразливості - EternalBlue (Windows) і EternalRed (Linux) – за допомогою яких, поширювався сумнозвісний троянець-шифрувальник WannaCry і шкідливий майнер криптовалюта Monero EternalMiner.

Приблизний розподіл заражених IoT-пристроїв, з IP-адрес яких були атаковані пастки спеціалістів лабораторії Касперського в другому кварталі 2018 року наведений в Таблиця 1.6.

Як можна помітити, з великим відривом попереду пристрої компанії MikroTik, що працюють під управлінням RouterOS. Причина, по всій видимості, в уразливості Chimay-Red.

Таблиця 1.6 – Розподіл заражених IoT-пристроїв

Пристрій	% заражених пристроїв
MikroTik	37,23 %
TP-Link	9,07 %
SonicWall	3,74 %
AV tech	3,17 %
Vigor	3,15 %
Ubiquiti	2,80 %
D-Link	2,49 %
Cisco	1,40 %
AirTies	1,25 %
Cyberoam	1,13 %
HikVision	1,11 %
ZTE	0,88 %
Unspecified device	0,68 %
Unknown DVR	32 %

Досить популярні атаки на сервіс віддаленого адміністрування пристроїв (специфікація TR-069), який працює на порту 7547. За даними Shodan, в світі налічується більше 40 мільйонів пристроїв, у яких цей порт відкритий. І це незважаючи на те, що ще не так давно вразливість стала причиною зараження мільйона роутерів Deutsche Telekom, а також «допомогла» поширенню шкідників родин Mirai і Hajime.

Ще один тип атак експлуатує уразливість Chimay-Red в роутерах MikroTik під керуванням RouterOS версії нижче 6.38.4. У березні 2018 року за її допомогою активно поширювався Hajime.

Зловмисники не обійшли стороною і IP-камери: в березні 2017 року ПЗ пристроїв GoAhead було виявлено кілька серйозних вразливостей, а через місяць після публікації інформації про це з'явилися нові модифікації троянців Gafgyt і Persirai, що експлуатують ці уразливості. Уже через тиждень після початку активного поширення цих шкідників число заражених пристроїв зросла до 57 тисяч.

8 червня 2018 року було опубліковано proof-of-concept до уразливості CVE-2018-10088 веб-сервера XionMai uc-httpd, який використовується в деяких «розумних» пристроях китайських виробників (наприклад, відеореєстратор KKMoon DVR). Вже на наступний день зареєстрована кількість спроб виявити пристрої, що використовують даний веб-сервер, зросла більш ніж в 3 рази. Винуватцем такого сплеску активності став троянець Satori, який раніше атакував роутери GPON.

1.2.3 Методи та цілі використання компрометованих пристроїв IoT

Головним завданням, яке зловмисники вирішують за допомогою IoT-шкідників, була і залишається організація DDoS-атак. Заражені «розумні» пристрої стають частиною ботнету, який по команді починає атакувати вказану адресу, позбавляючи хост можливості нормально обробляти запити реальних користувачів. Такими атаками, наприклад, продовжують займатися троянці сімейства Mirai і його клони, зокрема, зловредів Hajime.

Це найбільш безневинний сценарій для кінцевого користувача. Максимум, що загрожує власникові зараженого пристрою (і це дуже малоймовірний сценарій) - блокування з боку інтернет-провайдера. А «вилікувати» пристрій найчастіше можна за допомогою простого перезавантаження.

Інший тип корисного навантаження пов'язаний з криптовалюта. Наприклад, IoT-зловредів можуть встановлювати майнер на заражене пристрій. Але з огляду на невелику обчислювальної потужності «розумних» пристроїв, доціль-

ність такої атаки залишається під питанням, навіть незважаючи на їх потенційно велика кількість.

Більш хитрий і дієвий спосіб збагатитися на одну-дві кріптомонети придумали автори троянця Satori. IoT-пристрій виступає в їх сценарії в ролі своєрідного «ключа», що відкриває доступ до високопродуктивної ЕОМ:

- На першому етапі зловмисники намагаються заразити якомога більше роутерів, використовуючи відомі уразливості, а саме:
  - CVE-2014-8361 - RCE в miniigd SOAP service в Realtek SDK;
  - CVE 2017-17215 - RCE в прошивці роутерів Huawei HG532;
  - CVE-2018-10561, CVE-2018-10562 - обхід авторизації і можливість виконати довільні команди на роутерах Dasan GPON;
  - CVE-2018-10088 - переповнення буфера в XiongMai uc-httpd 1.0.0, який використовується в прошивках деяких роутерів та інших «розумних» пристроїв деяких китайських виробників.
- Використовуючи скомпрометовані роутери і вразливість CVE-2018-1000049 в механізмі віддаленого адміністрування ПЗ для Майнінга криптовалюти Etherium - Claymore, заміняючи адресу гаманця на свій.

Виявлений в травні 2018 року троянець VPNFilter переслідує інші цілі. Головна серед них - перехоплення трафіку зараженого пристрою, витяг з нього важливих даних (логіни, паролі тощо) і відправка їх на сервер зловмисників. Ось основні особливості VPNFilter:

- Модульна архітектура. Автори шкідника можуть «обладнувати» його новими функціями «на льоту». Так, на початку червня 2018 був виявлений новий модуль, який умів інжектувати javascript-код в перехоплені веб-сторінки.
- Стійкість до перезавантаження пристрою. Троянець прописує себе в стандартний для Linux-планувальник crontab, а також може змінювати конфігураційні параметри в незалежній пам'яті (NVRAM) пристрою.

- Використання TOR для комунікації зі своїм сервером управління.
- Можливість самознищення і виведення пристрою з ладу. Отримавши відповідну команду, троянець видаляє себе, а також перезаписує сміттєвими даними критичну частину прошивки після чого перезавантажує пристрій.

Спосіб поширення троянця досі залишається невідомим: його код не містить жодних механізмів самопоширення. Однак, можна припустити, що зазвичай, для зараження використовуються відомі уразливості в ПЗ пристроїв.

У найпершому звіті про VPNFilter говорилося про 500 тисяч заражених пристроїв. З тих пір їх стало більше, а список виробників вразливих гаджетів значно збільшився. На середину червня він включав пристрої таких брендів:

- ASUS;
- D-Link;
- Huawei;
- Linksys;
- MikroTik;
- Netgear;
- QNAP;
- TP-Link;
- Ubiquiti;
- Upvel;
- ZTE.

Ускладнює ситуацію ще й те, що пристрої цих виробників використовуються не тільки в корпоративних мережах, але часто і в якості домашніх роутерів звичайних користувачів.

1.3 Аналіз ботнет

Ботнет - це низка пристроїв, підключених до Інтернету, таких як комп'ютери, смартфони або пристрої IoT, безпека яких була порушена та контрольована передача стороннім особам. Кожен такий скомпрометований пристрій, відомий як "бот", створюється, коли шкідник проникає у девайс через програмне забезпечення. Контролер ботнету здатний керувати діями цих компрометованих комп'ютерів за допомогою каналів зв'язку, створених стандартними мережевими протоколами, такими як IRC та протокол передачі гіпертексту (HTTP). Архітектура ботнету еволюціонувала з часом, щоб уникнути виявлення та збоїв. Традиційно – ботові програми будуються як клієнти, які спілкуються через існуючі сервери. Це дозволяє оператору ботів (людині, що керує ботнетом) виконувати всі дії з віддаленого місця, що ускладнює пошук витоку трафіка. Більшість останніх ботнетів покладаються на існуючі peer-to-peer мережі для спілкування. Ці P2P ботові програми виконують ті самі дії, що й моделі клієнт-сервер, але вони не потребують центрального сервера для комунікації.

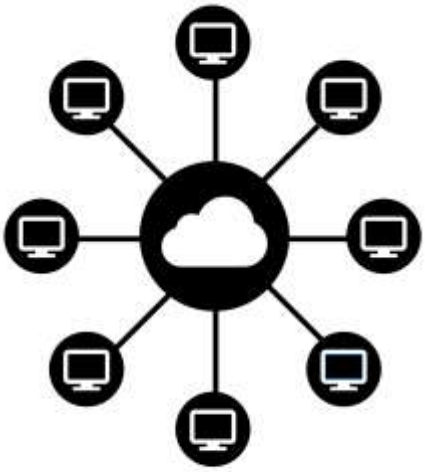
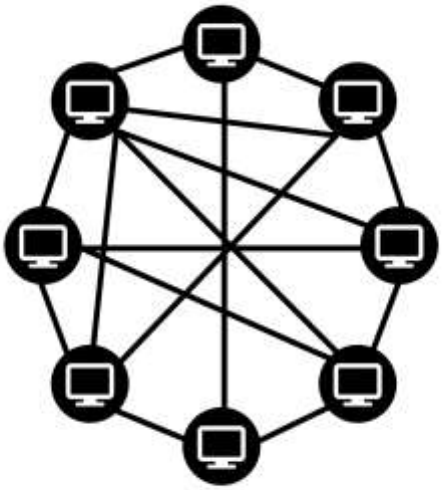


Рисунок 1.2 - мережа, створена на моделі клієнт-сервер, де окремі клієнти посилають запит до централізованих серверів

Перші ботнети в Інтернеті використовували модель клієнт-сервер (Рисунок 1.2) для роботи. Як правило, ці ботнети працювали у Internet Relay Chat мережах, доменах або веб-сайтах. Заражені клієнти отримували доступ до заздалегідь визначеного місця та чекали вхідних команд із сервера. Оператор ботів надсилає команди на сервер, який передавав їх клієнтам. Клієнти виконували команди та звітували про результати їх повернення слідкуючому за ботами. У випадку з ботнетами IRC, заражені клієнти підключаються до зараженого IRC-сервера та приєднуються до каналу, створеного для комунікації клієнт-клієнт. Оператор ботів надсилає команди до каналу через сервер IRC. Кожен клієнт отримує команди та виконує їх. Клієнти відсилають повідомлення до каналу IRC з результатами своїх дій.

У відповідь на зусилля, спрямовані на виявлення та знищення ботнетів IRC, зловмисники почали запускати шкідливе програмне забезпечення в одноранговых мережах. Ці боти можуть використовувати цифрові підписи, так що лише той, хто володіє приватним ключем – може контролювати ботнет. Наприклад ботнети Gameover ZeuS і ZeroAccess.

Новіші ботнети повністю працюють у P2P мережах. Замість того, щоб спілкуватися з централізованим сервером, P2P боти виконують ролі як сервера, який надає команди, так і клієнта, який їх отримує (Рисунок 1.3). Це дозволяє уникнути збоїв при падінні точки управління, що є проблемою для централізованих ботнетів. Щоб знайти інший інфікований пристрій, бот обережно аналізує випадкові IP-адреси, доки не зв'яжеться з собіподібним. Зв'язаний бот відповідає інформацією – версією програми та списком відомих ботів. Якщо



одна з

Рисунок 1.3 – одноранговова мережа (P2P), в якій взаємопов'язані вузли («peers»), які розподіляють ресурси між собою без використання централізованої адміністративної системи

версій ботів нижча, ніж інша, вони будуть ініціювати передачу файлів для оновлення. Таким чином, кожен бот збільшує свій список заражених машин і оновлює себе, періодично спілкуючись з усіма відомими ботами.

Цей приклад ілюструє, як ботнет створюється та використовується зловмисником:

- Хакер купляє або створює Trojan або exploit та використовує його для зараження комп'ютерів користувачів шкідливою програмою - ботом.
- Бот наказує інфікованому ПК підключитися до певного сервера керування (C&C - comand and control). Це дозволяє керівнику ботнета вести облік кількості активних ботів
- Потім керівник ботнету може використовувати ботів щоб збирати особисту інформацію користувачів або скористатися зчитуванням форм вводу, щоб викрасти особисті облікові дані, а також може продавати ботнет як DDoS або спам службу або ж продати облікові дані користувачів.
- Залежно від якості та можливостей ботів, їх цінність збільшується або зменшується.

Новіші боти можуть автоматично сканувати своє середовище та розповсюджувати себе за допомогою вразливостей та слабких паролів. Взагалі, чим більше вразливостей бот може сканувати та розповсюджувати, тим ціннішим він стає спільноті контролера ботнету.

1.3.1 Дослідження причин поширення ботнет

Зараз проблема погіршується через розповсюдження дешевих пристроїв, які належать до інтернету речей. Оскільки ці пристрої, як правило, мають незначну захищеність, компометація цих пристроїв не забирає великої кількості ресурсів. Це робить створення великих ботнетів, які знищують не один сайт за раз, занадто простим та ефективним засобом досягнення своїх цілей для багатьох зловмисників.

У жовтні ботнет, що складався зі 100 тисяч скомпрометованих пристроїв, частково вимкнув постачальника інтернет-інфраструктури. Вимкнення провайдера Dyn призвело до появи каскаду наслідків, що врешті-решт призвело до тимчасового зникнення з Інтернету довгого списку високопрофільних веб-сайтів, у тому числі Twitter і Netflix. Що ще більш цікаво – ботнет, що напав на Dyn, був створений за допомогою загальнодоступної шкідливої програми Mirai, яка значною мірою автоматизує процес примноження заражених девайсів.

Найкращим захистом для всього Інтернету буде запуск лише безпечного програмного забезпечення, що не відбудеться найближчим часом. Пристрої інтернету речей не розроблені з пріритетом на безпеку і часто не можуть бути виправлені після того як попадають до кінцевого користувача. Наприклад девайси IoT, що були заражені ботнетом Mirai – будуть уразливими, поки їх власники не викинуть їх назавжди. Ботнети стануть більшими й потужнішими просто тому, що кількість вразливих пристроїв збільшиться у кілька разів на найближчі кілька років.

Ботнети використовуються для здійснення шахрайств з кліками. Шахрайство з кліками – це схема, яка допомагає рекламодавцям думати, що люди натискають або переглядають їх інтернет-об’яви. Існує безліч способів здійснення шахрайств з кліками, однак, найімовірніше, зловмисник може вставити оголошення Google на власну веб-сторінку. Реклами Google платять власнику сайту відповідно до кількості людей, які на них натискають. Зловмисник наказує всім комп'ютерам свого ботнету неодноразово відвідувати веб-сторінку та натискати оголошення. Якщо розробники ботнет виявлять більш ефективні способи викупу доходів від великих інтернет-компаній, наше покоління буде спостерігачами того, як вся рекламна модель Інтернету буде зруйнована.

Так само, ботнети можуть використовуватися для уникнення спам-фільтрів, які працюють, частково знаючи які комп'ютери відправляють мільйони повідомлень електронної пошти. Вони можуть: прискорити вгадування паролей для взлому аккаунтів користувачів, майнити біткойни та робити будь-що,

що вимагає великої мережі комп'ютерів. Саме тому ботнети є великим бізнесом. Та кримінальні організації орендують час їх використання.

Найчастіше потрапляють у заголовки DDoS атаки за допомогою ботнетів. Фінансово-мотивовані групи використовують ці напади як форму вимагання. Політичні групи використовують їх для замовчування веб-сайтів, які їм не подобаються. Такі атаки, безумовно, будуть тактикою в будь-якій майбутній кібервійні.

Тож ботнет на основі інтернету речей так поширений через велику кількість вразливостей, які містять пристрої інтернету речей, відносно не оперативне оновлення протоколів безпеки задля закривання відомих вразливостей. Також чи не малу роль виконує велика кількість таких пристроїв.

1.4 Існуючі підходи до вирішення вразливостей на IoT пристроях

«Розумних» пристроїв стає все більше і, за прогнозами, до 2020 року їх кількість в кілька разів перевищить населення планети. Однак виробники все ще приділяють недостатньо уваги їх безпеці: немає нагадувань про необхідність зміни стандартних паролів при першому налаштуванні, немає повідомлень про появу нових версій прошивок, а сам процес оновлення може бути складний для звичайного користувача. Все це робить IoT-пристрої бажаною ціллю для зловмисників. Їх простіше заразити ніж персональний комп'ютер і при цьому вони можуть займати далеко не останнє місце в домашній інфраструктурі: одні керують усім інтернет-трафіком, інші можуть знімати відео, а треті керують іншими пристроями (наприклад, кліматична установка).

Шкідливе ПЗ для «розумних» пристроїв розвивається не тільки кількісно, але і якісно: в арсеналі зловмисників з'являється все більше експлойтів, які використовуються для самопоширення, а заражені пристрої використовуються для крадіжки персональних даних, видобутку криптовалюти, а не тільки для організації DDoS-атак.

Є декілька простих порад, які можуть мінімізувати ризик зараження пристроїв IoT:

- Закрити доступ із зовнішньої мережі до пристрою без крайньої необхідності;
- Періодичне перезавантаження допоможе позбутися від уже встановлених шкідників (але в більшості випадків ризик повторного зараження залишається);
- Регулярна перевірка наявності нових версій прошивки та оновлення пристрою;
- Використання складних паролей довжиною не менше 8 символів, що включають в себе букви різного регістра, цифри та спецсимволи;
- Зміна заводських паролей після першого запуску пристрою, під час першого налаштування (навіть якщо пристрій про це не просить);
- Закрити / заблокувати «зайві» порти, якщо є така можливість. Наприклад, якщо немає необходності підключатись до роутера по Telnet (порт tcp: 23), варто відключити його, щоб перекрити зловмисникам можливу лазівку.

Хоча ці поради й можуть частково закрити деякі існуючі проблеми та вразливості пристроїв інтернету речей, але це не вирішує проблему загалом, адже принцип розробки інтернету речей залишиться та нові вразливості будуть знайдені зловмисниками. Тож задача цієї роботи – це знайти можливість вирішити проблему безпеки IoT пристроїв системно. Це дозволить відрізати легкий шлях до компрометації таких пристроїв, що призведе до подорожчання продажі ботнет як послуги.

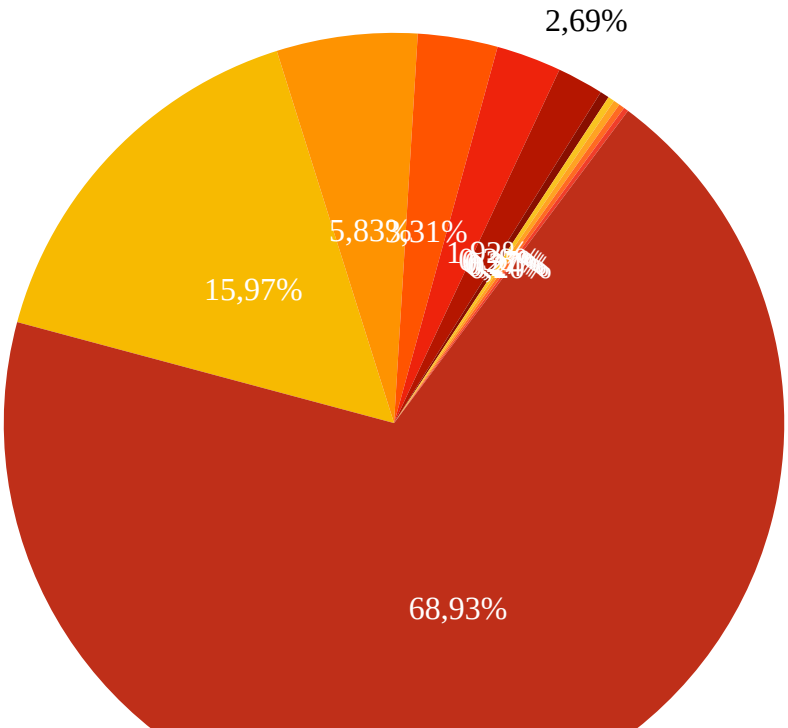
1.5 Висновок. Постановка задачі

Отримані результати аналізу IoT, основних вразливостей та методів їх використання вказують на те що:

- джерелом загроз для інтернету речей найчастіше є людина або група людей, вмотивована фінансово, політично або ж ідейно;
- вразливості, які використовуються зловмисниками найчастіше – спричинені низкою недоліків на етапі розробки IoT пристроїв;
- за останні декілька років більшість атак на IoT мали на меті заразити шкідником пристрій для послідуючого створення ботнет;
- з 2016 року при атаках на IoT пристрої більше 20% атак використовують ботнет Mirai та його модифікації (Рисунок 1.4);
- На даному етапі розвитку технології IoT – не існує міжнародного стандарту розробки пристроїв та систем інтернету речей, тож їх захищеність є не вирішеним питанням кібербезпеки.

Рисунок 1.4 – Розподіл популярності модифікацій Mirai

Таким чином, згідно приведених даних – необхідно провести глибокий



аналіз проблеми безпеки IoT, їх причини та знайти спосіб системно вирішити найбільш поширену та критичну проблему в системах інтернету речей – ботнет.

РОЗДІЛ 2. СПЕЦІАЛЬНА ЧАСТИНА

2.1 Аналіз методів реалізації ботнет

Щоб проаналізувати як ботнет діє та якими засобами реалізовується загалом – візьмемо найбільш поширений та доступний зразок для подальшого дослідження – Mirai. Такий вибір обумовлений тим, що Mirai – є зразковим прикладом дій ботнету, а також найбільш популярним серед зловмисників. За даними щоквартального дослідження Kaspersky Lab, станом на початок осені – більше 20% атак на IoT реалізовані за допомогою комплексу шкідливого ПЗ та ботнету Mirai або його модифікацій.

2.1.1 Аналіз Mirai

Хоча інші ботнети IoT, такі як BASHLITE та Carna, передували Mirai, він був першим, який вважається значною DDoS загрозою. Раптовий підйом і масивний масштаб Mirai є наслідком комбінації факторів – ефективне розповсюдження на основі сканування в Інтернеті, широкомасштабне використання різноманітних вразливостей у продуктах IoT та розуміння того, що збереження простої поведінки ботнету дозволить заразити багато пристроїв - все це відіграло певну роль. Дійсно, Mirai породив багато варіантів, які слідують за тією ж стратегією зараження, що приводить до припущення, що «ботнет IoT – нова норма DDoS атак».

Статистичні дані та глибокий аналіз роботи Mirai, що приведені нижче, взяті з різноманітних джерел збору інформації, включаючи мережеві телескопічні зонди, інтернет-сканування банерів, IoT ханіпоти, DNS сліди та журнали, надані жертвами нападу. Ці унікальні набори даних дозволили провести комплексний аналіз Mirai і передбачити технічні та нетехнічні засоби захисту, які можуть запобігти майбутнім атакам.

Відстежуючи спалах Mirai було виявлено, що ботнет заразив близько 65 000 пристроїв IoT протягом перших 20 годин, перш ніж досягти стабільного рівня у 200 000 - 300 000 заражень. Ці боти потрапили до вузької групи географічних регіонів та автономних систем, з Бразилією, Колумбією та В'єтнамом, на які непропорційно припадає 41,5% випадків інфікування. Підтверджено, що Mirai орієнтована на різноманітні IoT і вбудовані пристрої, починаючи від відеореєстраторів, IP-камер, маршрутизаторів та принтерів, але, виходячи з кінцевого складу Mirai – на нього сильно вплинули частки ринку та проектні рішення декількох виробників побутової електроніки.

Статистично аналізуючи більше тисячі зразків шкідливих програм, було задокументовано еволюцію Mirai на десятки варіантів, які поширюються багатьма конкуруючими операторами ботнетів. Ці варіанти намагалися покращити технології запобігання виявленню Mirai, додавати нові цілі IoT пристроїв та забезпечити додаткову DNS стійкість. Mirai застосував свої можливості до розвитку, щоб запустити понад 15 000 атак не тільки проти високопрофільних цілей (наприклад, Krebs on Security, OVH та Dyn), а також проти численних ігрових серверів, телекомунікаційних та анти-DDoS провайдерів, та проти інших, здавалося б, не причасних сайтів. Хоча здавалося, що Mirai зловживає DDoS атаками, майбутні штами зловмисного програмного забезпечення IoT можуть використовувати доступ до скомпрометованих маршрутизаторів для клікфроду, камери для вимагання або хмарного сховища для видобутку криптовалюти (Рисунок 2.1).

Рисунок 2.1 - хронологія Mirai - Великі атаки (червоний), вразливості (жовті) та події (чорні), пов'язані з ботнетом Mirai.

2.1.2 Алгоритм роботи Mirai

Mirai розповсюджується, спочатку входячи у швидку фазу сканування, де він асинхронно і "безвинно" передає зонди TCP SYN псевдовипадковим IPv4-адресам, за винятком тих, що містяться у hard-coded IP-чорному списку, на порту Telnet TCP 23 та 2323 (далі TCP / 23 та TCP / 2323). Якщо Mirai виявляє потенційну жертву, він входить до фази брутфорс логіну, коли він намагався встановити з'єднання Telnet, використовуючи 10 партій імені користувача та паролів, вибрані випадково з попередньо налаштованого списку з 62 облікових даних. Після першого успішного входу Mirai надсилає IP-адресу потерпілих та пов'язані з ним дані на hard-coded сервер звіту.

Окремі програми завантажувача асинхронно заражають ці вразливі пристрої шляхом входу в систему, визначаючи основне системне середовищи та, нарешті, завантажуючи та виконуючи зловмисне архітектурно-спеціалізоване програмне забезпечення. Після успішного зараження – Mirai намагається приховати свою присутність, видаливши завантажений бінарний файл і змінивши назву його процесу на псевдовипадковий ряд цифр. Як наслідок, зараження Mirai не зберігаються в процесі перезавантаження системи. Для того, щоб зміцнити себе, шкідлива програма додатково знищує інші процеси, пов'язані з TCP / 22 або TCP / 23, а також процеси, пов'язані з конкуруючими інфекціями, включаючи інші варіанти Mirai, .anime і Qbot. На цьому етапі бот слухає команди атаки з C&C сервера (C&C), одночасно сканує, у пошуку нових жертв (Рисунок 2.2).

2.1.3 Активне сканування пристроїв, заражених Mirai

У першому розділі були розглянуті результати дослідження Лабораторії Касперського щодо заражених пристроїв. Для того щоб дослідити та проаналізувати деталі роботи маскуючого ПЗ в Mirai – необхідно провести дослідження та сканування на наявність таких пристроїв в мережі. Незважаючи на

те, що Mirai найчастіше розглядається як ботнет IoT, аналізу заражених пристроїв за весь час роботи ботнету було дуже мало. Щоб визначити модель та виробника пристроїв, заражених Mirai, було вирішено використовувати Censys, який активно сканує простір IPv4 і агрегує дані про хости в Інтернеті. Аналіз був зосереджений на скануваннях таких протоколів як: HTTPS, FTP, SSH, Telnet та CWMP.

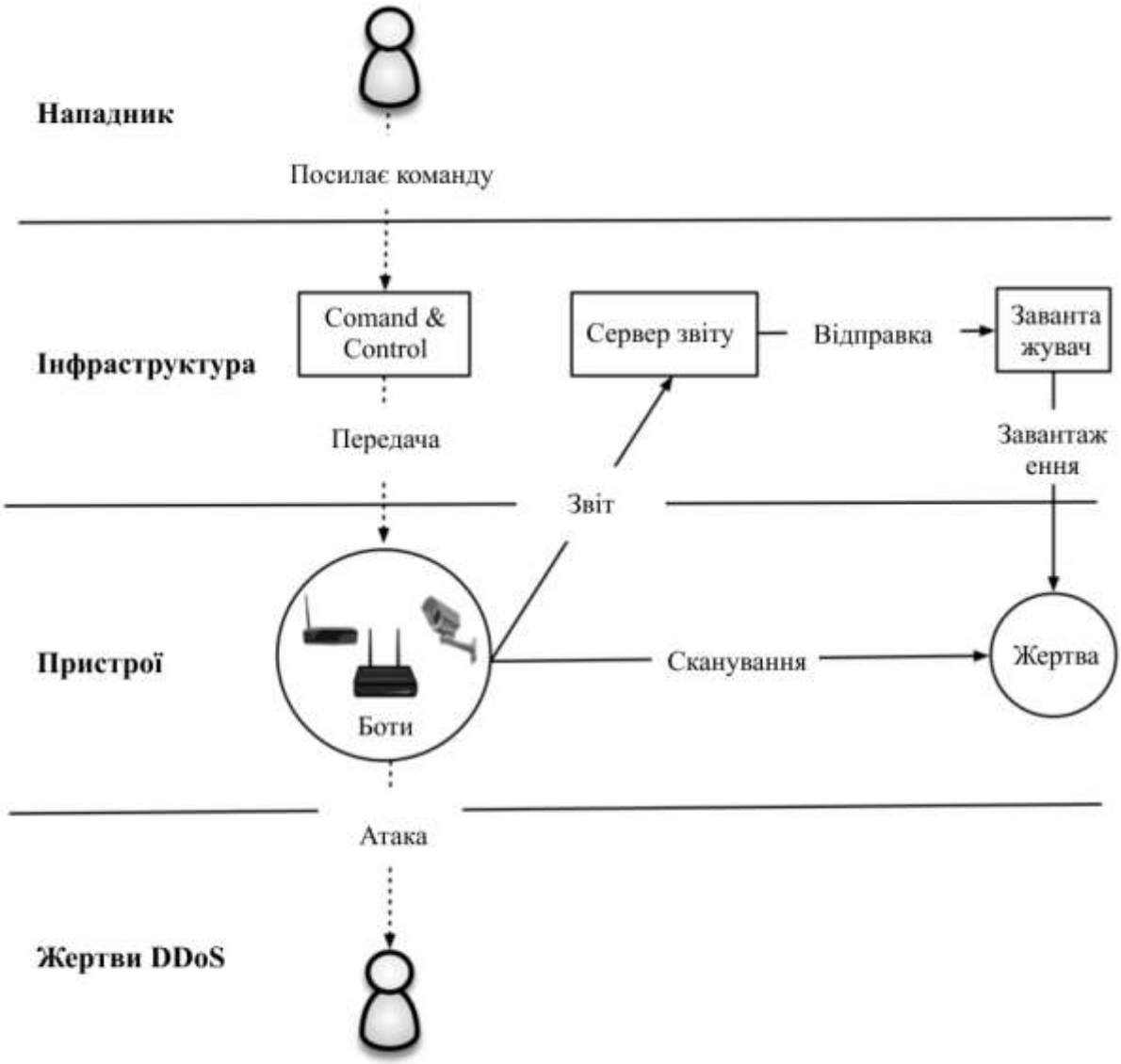


Рисунок 2.2 – Алгоритм роботи Mirai

Ряд причин ускладнює точне маркування пристрою. По-перше, Mirai після зараження негайно вимикає загальні служби, до яких звертаються ззовні (наприклад, HTTP), що перешкоджає скануванню заражених пристроїв. По-друге, сканування Censys зазвичай займає більше 24 годин, протягом якого пристрої можуть присвоїти нові IP-адреси. Також, Censys виконує сканування різних протоколів в різні дні, що ускладнює збільшення специфіки мітки, об'єднуючи банери кількох служб. Було вирішено пройти ці обмеження, обмеживши аналіз банерами, зібраними протягом двадцяти хвилин роботи сканування (період часу, після якого закінчується сканування). Це невелике вікно пом'якшує ризик помилкового зв'язку банерних даних неінфікованих пристроїв з інфікованими Mirai через DHCP.

У постфільтруванні, набір даних включав 1,8 мільйона банерів, пов'язаних з 1,2 мільйонами інфікованих Mirai IP-адрес (Таблиця 3.1). Для CWMP було більше зразків, а найменше для SSH. Пристрої з відкритими службами, які не закриті Mirai (наприклад, HTTPS та FTP), можуть повторюватися в базі даних Censys під час нашого вікна вимірювання (через припинення) і, таким чином, призводять до надмірного підрахунку при порівнянні за протоколами. Таким чином, було вирішено досліджувати протоколи окремо один від одного і обмежуватися вимірами, які враховують лише відносні пропорції, а не абсолютний розрахунок заражених хостів.

Наприкінці, необхідно було обробити кожний банер зараженого пристрою для ідентифікації виробника пристрою та моделі. Спочатку була застосована множина регулярних виразів, що використовувалася зондами служби Nmap для відбитків пальців. Nmap успішно обробив 98% банерів SSH та 81% банерів FTP, але відповідав лише 7,8% банерам Telnet. З метою збільшення охоплення, а також включення HTTPS та CWMP (які у Nmap відсутні зонди), були побудовані регулярні вирази для створення банерів для виробників пристроїв та моделей. Після цього – було виявлено, що в багатьох випадках не було достатньо даних для ідентифікації моделі та виробника з використанням банерів FTP, Telnet, CWMP та SSH, а відбитки пальців Nmap містять лише за-

гальні описи. Загалом було визначено тип пристрою, модель та виробника 31,5% банерів (Таблиця3.2).

Таблиця 2.1 – Був визначений тип пристрою, модель та виробник для 31,5% активних банерів сканування. Банери протоколів різко варіювалися в можливості ідентифікації пристрою по ним; сертифікати HTTPS є найбільш змістовними носіями, а SSH - найменш змістовними

Служба	Банери	Пристрої	Ідентифіковано
HTTPS	342,015	271,471	79,4 %
FTP	318,688	144,322	45,1 %
Telnet	472,725	103,924	22,0 %
CWMP	505,977	35,163	7,0 %
SSH	148,640	8,107	5,5 %
Загалом	1,788,045	587,743	31,5 %

Ці дані показують, що ідентифікувати модель та виробника пристрою стає складнішим або легшим в залежності від протоколу передачі даних, який використовується. Це наводить на висновок, що навіть коли відомо про вразливість окремих моделей пристроїв або версій прошивки, виявлення таких пристроїв у мережі може бути надзвичайно важким. Це зробило дослідження складним завданням, але це, також, ускладнює операторам мережі виявити вразливості на своїх пристроях або пристроях своїх користувачів для їх виправлення.

2.2 Визначення типів компрометованих IoT пристроїв

Хоча поверхневі докази показали, що Mirai націлений на пристрої IoT, словник імен користувача та паролів Mirai, за замовчуванням, включає в себе маршрутизатори, відеореєстратори та камери, а його джерело складається з декількох вбудованих апаратних конфігурацій – далі пропонується поглиблений аналіз обох цільових пристроїв та аргументований їхніми успішними зараженнями.

Щоб зрозуміти типи пристроїв, на які націлений Mirai, був проаналізований список облікових даних, закодованих в бінарні файли, які були зібрані в ході аналізу. Загальнуа кількість паролей становила – 371, а через ручну перевірку було виявлено 84 пристрої та / або постачальника, пов'язані з цими паролями. Багато паролів були занадто загальними для зв'язування з певним пристроєм (наприклад, "password" застосовується до пристроїв великої кількості виробників), тоді як інші лише надавали інформацію про базові програми (наприклад, "postgres"), а не пов'язані пристрої . Пристрої, які були ідентифіковані – в більшості були мережевими пристроями для зберігання даних, домашніми маршрутизаторами, відеокамерами, відеореєстраторами, принтерами та телевізійними приймачами, створеними десятками різних виробників.

Заплановані цілі Mirai не обов'язково відображають розбиття інфікованих пристроїв. Для аналізу були використані банерні пристрої, зібрані компанією Censys, для визначення моделей та виробників заражених пристроїв. Результати за всіма п'ятьма протоколами вказують на те, що камери безпеки, відеореєстратори та споживачі маршрутизаторів представляють більшість в мережі Mirai . (Таблиця 2.3). Виробники, відповідальні за найбільш інфіковані пристрої, які можливо ідентифікувати: Dahua, Huawei, ZTE, Cisco, ZyXEL і MikroTik.

Ціль	CWMP (28.30%)	Telnet (26.44%)	HTTPS (19.13%)	FTP (17.82%)	SSH (8.31%)
Роутер	4,7 %	17,4 %	6,3 %	49,5 %	4,0 %
Ціль	CWMP (28.30%)	Telnet (26.44%)	HTTPS (19.13%)	FTP (17.82%)	SSH (8.31%)

Камера / DVR	-	9,4 %	36,8 %	0,4 %	-
Сховище	-	-	0,2 %	-	0,2 %
Медіа-пристрій	-	-	-	0,1 %	-
Файервол	-	-	0,1 %	-	0,2 %
Безпека	-	-	-	-	0,1 %
Інші	0,0 %	0,1 %	0,2 %	0,0 %	0,0 %
Невідомо	95,3 %	73,1 %	56,4 %	49,0 %	95,6 %

Таблиця 2.3 – Найпоширеніші типи пристроїв Mirai.

Зазначимо, що ці результати відхиляються від початкових звітів засобів масової інформації що Mirai переважно складається з відеореєстраторів та камер. Ймовірно, це пов'язано з еволюцією зловмисного програмного забезпечення Mirai з часом, що змінило склад інфікованих пристроїв. Дивлячись на поздовжню кореляцію Пірсона з виробниками найпоширеніших пристроїв, можна спостерігати скромну стійкість, за винятком двох періодів подій: фази швидкого зростання в середині вересня 2016 року та настання CWMP у кінці листопада 2016 року (Рисунок 2.3). Під час швидкого зростання поява користувацьких маршрутизаторів, вироблених ASUS, Netgear і Zhone, витіснила маршрутизатори D-Link та Controlbr DVR у провідних 20 пристроях. Dahua, Huawei, ZyXEL та ZTE послідовно залишалися в топ 20.

Отримані дані вказують на те, що деякі з провідних світових виробників побутової електроніки не мали достатньої практики безпеки, щоб пом'якшити такі загрози, як Mirai, і ці виробники відіграватимуть ключову роль у формуванні вразливості.

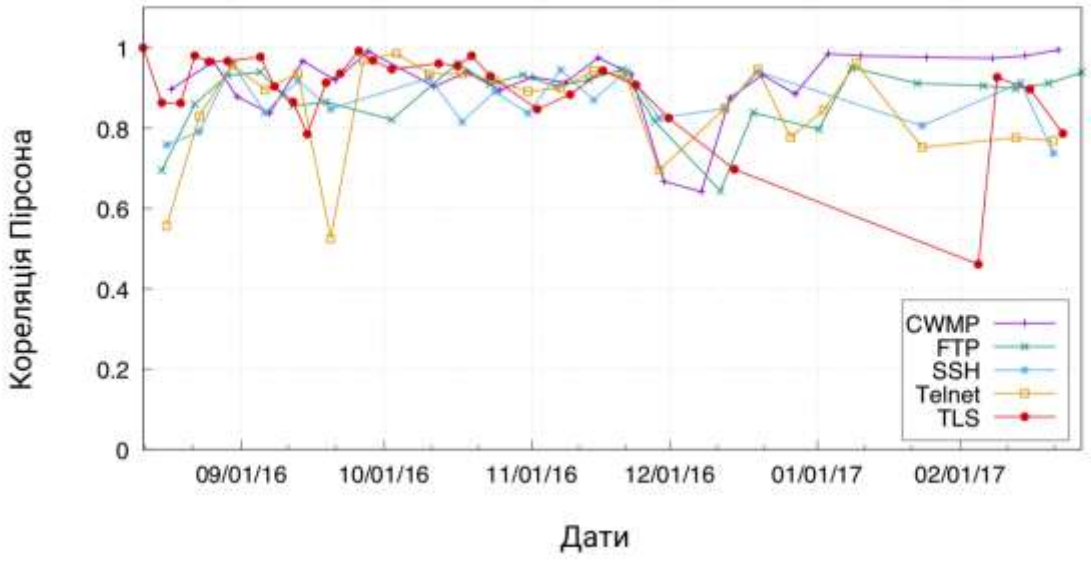


Рисунок 2.3 – Стабільність пристроїв з плином часу

Mirai привернув увагу технічних та регуляторних викликів, пов'язаних із забезпеченням захисту, керованих споживачами, безінтерфейсних пристроїв IoT. Зловмисники користуються перевагами переворотом тенденцій безпеки в останні два десятиліття, особливо поширеними на пристроях IoT. На відміну від настільних і мобільних систем, де невелика кількість безпеко-орієнтованих вендорів контролює найбільш чутливі частини програмного забезпечення (наприклад, Windows, iOS, Android) – пристрої IoT більш неоднорідні та виробники питаннями безпеки переважно нехтують. Шукаючи відповідні технічні та політичні заходи для сучасної екосистеми IoT, можна спиратися на досвід роботи з вірусами для настільних комп’ютерів з 2000-х років.

2.3 Аналіз існуючих методів протидії ботнет

Так як ботнети існують вже більше 20 років, то й методів протидії їм вже з'явилось чимало. Для аналізу їх ефективності у динамічній сфері IoT – було вирішено взяти для дослідження два основні типи підходів до протидії ботнетам: класичний та проактивний. На основі отриманих результатів аналізу – буде оцінено можливість використання цих методів, або їх частин, у подальшій роботі над проектуванням системи безпеки інтернет речей у розрізі теми їх захисту від ботнет.

2.3.1 Класичні методи протидії

Традиційні способи боротьби з ботнет зазвичай обмежуються виявленням центральної слабкої точки в їх інфраструктурі, для маніпуляції, порушення або блокування. Найпоширеніший спосіб полягає в тому, щоб співпрацювати з провайдером Інтернет-послуг для отримання доступу та вимкнення центрального компонента, що призвело б до втрати контролю ботнетом власникиком: так як ботнет він більше не зможе наказувати. Такі дії часто виконуються під час надзвичайної реакції на інцидент, який вже відбувається, наприклад – атака DDoS. Хоча цей курс дій виявився ефективним (наприклад, припинення роботи сервера C&C на базі IRC забороняє ботам отримувати команди та машини, які вже беруть участь у нападі рано чи пізно перезавантажуються), але він вимагає доступу до ПК, та, що найголовніше, бажання співпрацювати в відповідальній установі.

Класичні контрзаходи проти ботнетів мають три різні точки для атаки:

1. Сервер керування (C&C);
2. Ботнет-трафік;
3. Заражені комп'ютери;

Далі буде пояснення контрзаходів з їх можливостями та труднощами. Мета полягає в тому, щоб показати їх відмінності і в чому відмінність від проактивного способу.

2.3.1.1 Відключення серверу C&C

Найбільш перспективним підходом є видалення бази ботнету, якою є сервер C&C. Вимкнення C&C хоста дозволяє вивести з ладу весь ботнет за один раз. На жаль, це можливо лише за умови виконання всіх наступних умов:

1. Ботнет використовує централізовану структуру;

2. Розташування сервера C&C відоме;

3. Провайдер співпрацює;

Якщо не виконано хоча б одне з цих умов, сервер C&C не може бути видалений. Нові ботнети більше не покладаються на централізовану структуру. Замість цього вони використовують функцію однорангової мережі (P2P) або multiproxy структури, щоб приховати своє центральне походження. Таким чином, маловірогідно можна знайти розташування C&C сервера. Якщо використовуються декілька фіксованих серверів – всі вони повинні бути вилучені. Коли відомо про місцезнаходження, провайдер, який розміщує сервер C&C, повинен співпрацювати. Дуже часто ботнети контролюються з розташувань, розміщених на так званих *кулестійких хостерах*, які не реагують на запити про зловживання або, навіть гірше, переміщують сервер до дочірніх партнерських компаній, до тих пір, пока не спаде тиск на хостінг. Правоохоронні процедури часто залишаються одним кроком, коли хостингові служби раптово переходять до дочірнього постачальника. Різні організації, які відстежують атаки в Інтернеті, отримують так багато натяків на можливі сервери C&C, що вони не можуть обробляти, стежити та перевіряти дії проти кожного запиту про C&C окремо. Кількість умов є частиною проблеми, що існує така велика кількість ботнетів, і вона все більше зростає.

При цьому, виведення серверів C&C з ладу не завжди схоже на видалення кореня ботнету. Заражені комп'ютери також можуть містити функціональні можливості для автономного розповсюдження, а також іншу логіку відкату, яка виконується у разі відключення C&C. Це створює додатковий трафік і може призвести до послідуючого масштабного зараження обчислювальних пристроїв.

У деяких випадках відомо де було здійснено захоплення ботнету, з метою видати команди, які роблять боти зупиняють атаку або деінсталюють себе. Хоча цей підхід є делікатним стосовно відповідальності за наслідки, спричинені інфікованими машинами, він вкрай успішний одночасно. Напади зупинені негайно і в той же час – ботнет закривається остаточно, без можливості повернути його. Проте, шанси, що такі дії будуть успішними, в більшості випадків залежить від можливості співпраці з відповідальними постачальниками інфраструктури.

2.3.1.2 Розмивання зловмисного трафіку

Якщо сервер C & C не може бути вимкнений, іншим варіантом може бути переспрямування зловмисного трафіку на так звані *свердловини*, стратегія, яка увійшла до останніх методів пом'якшення. Свердловини записують шкідливий трафік, аналізують його, а потім перенаправляють його таким чином, що він не може досягти первісної цілі, для якої вона призначена. Одним з прикладів використання свердловин є DDoS *нуль-маршрутизація*. У випадку, якщо спостерігається трафік, що належить до спроби реалізації DDoS, він скидується, а іноді підраховується для подальшого аналізу. Нуль-маршрутизація DDoS на кордонних-маршрутизаторах є перспективним підходом для пом'якшення атак DDoS, але тут з’являються проблеми з надійністю ідентифікації трафіку, пов'язаного з атакою, та чистим розчленуванням потоків даних високої пропускної спроможності на ранній стадії. Це, як правило, можливо лише на рівні інтернет-

провайдера. Інший варіант - спільна всесвітня ініціатива між провайдерами, але це, очевидно, поза всяким питанням.

2.3.1.3 Очищення заражених систем

Найбільш стійким контрзаходом проти ботнетів є очищення всіх заражених систем та видалення встановлених ботів. Хоча це усуває повну потужність ботнету, це також є найбільш складним і найбільш важким в управлінні методом протидії. На сьогоднішній день власники або адміністратори несуть відповідальність за те, щоб їхні системи були чисті від інфекцій. Йдеться лише про рекомендації та технічну консультацію. Оскільки більшість користувачів навіть не знають про зараження своєї ЕОМ, не кажучи вже про можливість видалення зловмисного програмного забезпечення, тож глобальне очищення неможливе. Величезні рекламні кампанії про Conficker та кількість ще заражених систем показують, що навіть при інтенсивних попередженнях – чекати від простих користувачів повного очищення не варто.

Стандартна рекомендація щодо захисту систем від ботнет полягає в тому, щоб використовувати брандмауери та сучасне антивірусне програмне забезпечення (AV). Брандмауерами є профілактичною функцією, яка в багатьох випадках блокує атаки ззовні. Зростаюча кількість вразливостей drive-by-exploits, використовує помилки в браузері користувача для зараження системи та мобільність шкідливих даних на ноутбуках або USB-накопичувачах, відкриває цілий ряд нових векторів зараження, які обходять брандмауери. Антивірусне програмне забезпечення – реактивна функція. Перш ніж він зможе виявляти щонебудь, підписи мають бути доступними, і шкідливі дані повинні бути на цільовому комп'ютері. Якщо підписів поки що не існує, то систему не можна захистити. Тести різних AV-баз показали, що деякі показники виявлення становлять менше ніж 80%. Після зараження системи бот може розповсюджувати та виконувати шкідливі дії, доки AV-підписи не стануть доступні і зможуть бути використані. Часто AV-бази застаріли і не оновлюються на регулярній основі. Крім того, різні боти вимикають AV-сканери або ховаються такими шляхи, які неможливо виявити звичайними сканерами.

Загалом, глобальна очистка, яка потрібна щоб ефективно відняти владу у ботнетів, виглядає нездійсненною.

2.3.1.4 Висновок про класичні стратегії

Аналіз в цьому розділі показує, що на сьогоднішній день рівень успішності контрзаходів ботнету залежить в основному від організаційних та політичних загальних умов. З огляду на те, що налагодження співпраці або дипломатичних угод вимагає часу, доходимо до висновку, що встановлення відповідних відносин, що легітимує співпрацю для спільних дій, не підходить як спеціальна схема боротьби з поточними нападами.

Ситуація посилюється, враховуючи, що сучасні інфраструктури ботнету не підпадають під відповідальність одного суб'єкта. Натомість, розподілені однорангові мережі працюють у всьому світі, тому вимикання локальних частин (часто не більше, ніж одиночні ЕОМ) не буде ефективним рішенням. В цілому, контрзаходи, які потребують тісної співпраці, сьогодні, як правило, є нездійсненними як з технічних, так і з політичних причин.

У минулому були дискусії, в яких експерти заявили, що припинення роботи серверів C&C стає марним, оскільки вони майже завжди будуть замінені новими, більш захищеними системами. Ця прискорена гонка озброєнь в кінцевому рахунку призведе до складної технології ботнету швидше, ніж без пом'якшення наслідків. Позитивний підхід залишає потенційну цільову зону наодинці з існуючою загрозою. Врешті-решт, обмеження методів пом'якшення наслідків для уникнення або блокування поточних атак - це визнання безсилля. Доречним буде запропонувати поєднання класичних методів з додатковими проактивними стратегіями.

2.3.2 Проактивні заходи та стратегії

Класичні контрзаходи є дуже хорошими кроками для пом'якшення сили ботнет, але останні події показують, що вони підходять лише до певної міри. Новіші ботнети використовують більш складні технології, що заперечують використання класичних контрзаходів через труднощі, описані в попередньому

розділі. Незважаючи на те, що нові структури, представлені останніми ботнетами, ускладнюють застосування класичних контрзаходів, вони відкриті для більш агресивної контр-тактики. Цей розділ пояснює загальні принципи, які можуть бути використані для створення наступальних контрзаходів проти ботнет та зосереджений на технічних можливостях.

Дослідження структури ботнет часто є першим кроком для пошуку початкової точки для можливих контрзаходів. Характерною властивістю всіх ботнет є те, що вони повинні дозволяти новим машинам, які працюють на ненадійних платформах, приєднатися до мережі. Це важливий аспект для підходів до контрзаходів: не обмежуватися лише діяльністю ззовні – можна приєднатися до мережі, виконувати дослідження, будучи частиною інфраструктури самостійно, і навіть можна вміщувати ботнет або зруйнувати його зсередини . Крім того, боти поширюються, щоб заразити більше систем і збільшувати мережу. Зразки шкідливого програмного забезпечення, які важко отримати, можуть бути проаналізовані (напр. за допомогою реверс-інжинірінгу), щоб дізнатися про їх внутрішні компоненти. Дізнавшись про функціональні особливості бота, часто можна створити фальшивого, який стане частиною ботнета, щоб спостерігати або перешкоджати внутрішньому зв'язку. Ця процедура завжди можлива, тому що вся інформація про початкове завантаження повинна бути включена в бінарні файли шкідливого ПЗ і, отже, може бути клонована.

Наступальні стратегії можна розділити на три різні категорії: пом'якшення наслідків, маніпулювання та експлуатація. Ступінь можливої відповідної дії залежить в основному від топології, що використовується ботнетом. Зокрема, децентралізовані та рухомі топології можуть залишити лише декілька шансів на подібні контрзаходи.

Стратегії для пом'якшення наслідків є нападними технічними засобами, які уповільнюють роботу ботнетів за рахунок витрачання їх ресурсів. Прикладом можуть бути тимчасові спроби DoS на сервери C&C, створення з'єднань з зараженими машинами або блокування зловмисних доменів. Стратегії маніпулювання використовують командний рівень. Знання про командні протоколи

мають важливе значення для маніпулювання та введення команд. Необхідні знання про протоколи включають використання криптографії. Незважаючи на те, що криптографія може повністю заперечувати перевірку обміну даними про ботнети, але приклад дослідження Waledac показує, як цього можна досягти, навіть якщо використовуються криптографічні методи, такі як RSA та AES. Можливими маніпуляціями можуть бути зміна або видалення команд DDoS або спаму, а також команд для завантаження та виконання програм, що дозволяє віддалене очищення зараженого пристрою. Менш агресивні варіанти, ніж виконання програм на віддалених комп'ютерах, можуть полягати у вилученні зібраних особистих даних, таких як кредитні картки чи банківські реквізити, заміни їх підробленою інформацією або у команді, зупиняючою збір таких даних. Нарешті, експлуатація - це особлива стратегія, яка використовує помилки, знайдені в ботах. Подібно помилкам в інших продуктах, їх можна використовувати для виконання дій на заражених машинах. Незважаючи на те, що ця категорія є найпотужнішою, це найбільший ризик, оскільки вразливості можуть легко привести до пошкодження та навіть повної поломки систем та пристроїв.

Не кожна стратегія може бути застосована до кожного ботнету. Деякі з них значною мірою залежать від топології ботнету. Особливо децентралізовані ботнети пропонують ряд можливостей, які будуть описані та пояснені у цьому розділі.

2.3.2.1 Атака на адресний рівень

Обговорюючи стратегії, спрямовані на маршрутизацію та адресний рівень інфраструктури ботнету – важливо зрозуміти, що механізм маршрутизації, який використовується ботнетом, необхідний для адреси хостів або C&C серверів відповідно. Командний рівень, навпаки, працює на поверхні схеми адресації, щоб забезпечити комунікаційну мережу, яка накладається на взаємопов'язані пристрої.

Найпоширеніший спосіб, коли бот звертається до центрального сервера C&C, – це ім'я DNS, яке перенаправляє до IP-адреси – адресація відбувається у два етапи. Кожна фаза становить потенційну відправну точку для втручання. Наприклад, запити DNS зазвичай обробляються локальним DNS-resolver, який, у свою чергу, пересилає запит на авторитетний DNS-сервер. Цим локальним resolver'ом керує адміністратор сайту, і його легко запрограмувати, щоб повернути спеціально створену відповідь на конкретні запити. Те саме стосується і маршрутизації IP: локальні маршрутизатори можуть бути обладнані елементами таблиці маршрутизації для відключення певних адрес або перенаправлення їх на різні вузли (sinkholing - це термін для перенаправлення спроб підключення до спеціального сервера для ідентифікації заражених машин). Як наслідок, обидва кроки приводять до того, що боти в локальній мережі не можуть зв'язатися з оригінальним сервером C&C, і навіть можуть керуватися псевдосервером. Втручання, як описано вище, завжди вимагає стратегії «man in the middle". Проте, не завжди необхідно змінювати конфігурацію вбудованих пристроїв. Існують підходи, які демонструють живу модифікацію відповідного мережевого трафіку.

Сучасні ботнети використовують більш складні схеми адресації, які також працюють як накладна мережа на базі IP-Інтернету. Прикладами можуть слугувати одноранговi мережі. Вони мають свою власну схему адресації з метою збільшення гнучкості та децентралізації. Для проникнення в адресний рівень цих ботнет необхідна стратегічна позиція. Загальний підхід полягає у введенні ретельно відстеженого та керованого вузла, який є ідеальним клоном оригінально.

Навіть якщо C&C серверів не можна досягти фізично, вони повинні бути доступними через Інтернет, оскільки ботам потрібно зв'язатися з ними для отримання команд. Це може бути використано для послаблення ботнету шляхом створення DoS на сервер. І таким чином, контрольований союзником DDoS зробить сервер недоступним. Крім того, ботнети часто покладаються на технологію, яка схильна до слабкості до конкретних атак, наприклад, протокол

транспортного рівня TCP. Черга резервного копіювання TCP-сервера C&C може бути заповнена спробами з'єднання викликати умови відмови в обслуговуванні, перетворюючи зброю ботнету на себе. Це особливо корисно для більшості бот-серверів на базі HTTP, де встановлюються нові зв'язки для кожного командного запиту. Були оцінені різні комбінації служб і операційних систем та знайдена атака TCP DoS, якою можна легко керувати з дуже малимм ресурсами. Під час дослідження можна було достовірно зменшити ймовірність встановлення з'єднань із TCP-серверами до менш ніж 5% тільки з однією наступальною ОЕМ. Один хост може тримати чергу резервної копії служби жертви, блокуючи всі подальші спроби з'єднання і тим самим заважаючи ботам отримувати або надслати запити команд. Така операція може бути розроблена таким чином, що неможливо буде розрізнити спроби з'єднання з тими, що випускаються ботами. В результаті, будь-яка контр-дія, що має на меті заблокувати запити, також заблокує всі "законні" боти. Ці випробування показали, що одна ЕОМ може тримати службу TCP без відповіді, просто ініціюючи та завершуючи рукостискання у три сторони та підтримуючи з'єднання як можна довше. Така атака призводить до зменшення кількості ботів, здатних зв'язатися з сервером C&C і брати участь у зловмисних діях.

Ще одна подібна атака – це флуд посилання або мережі, де знаходиться сервер C&C з пакетами, які споживають всю доступну пропускну здатність. Проте, очевидно, така операція потребує більше ресурсів, тому що більше пакетів потрібно відправити. Атака віддзеркалення може бути використана для посилення обсягу відправленого трафіку, однак, це потребую підключення сторонніх ресурсів та, ймовірно, дозволу власників сайтів, які, очевидно, не будуть його надавати.

2.3.2.2 Атака на командний рівень

Напад на командний рівень ботнет вимагає знання протоколу, який використовується. Простим прикладом може бути мережа на базі IRC, де *команда*

видалення наказує ботам вилучати себе з заражених систем. Багато класичних ботів реалізують таку інструкцію. Впровадження команди вимагає або керування сервером C&C, або боти повинні бути перенаправлені на інший сервер, виконавши атаку на адресний рівень, який потім поширює інструкцію з видалення. Інші боти не мають можливості видалення, але пропонують функції оновлення, які можуть бути використані для заміни шкідливого програмного забезпечення на безпечну прошивку або програму, яка сканує та, врешті-решт, видаляє бот (подібно до антивірусного сканера).

У поєднанні з проникненням до адресного рівню стають можливими інші підходи: оригінальні команди можна прослуховувати, перехоплювати та модифікувати. Спеціальний протокол міг би здійснити перевірки, щоб зробити такі маніпуляції неможливими, однак подібні заходи ще не були виявлені в ботнетах.

Загалом, для того, щоб насправді проводити атаку на проникнення у ботнет, необхідна комбінація дій як з адресацією, так і з командним рівнем. Переадресація ботів на контрольований сервер або для знешкодження, або для того, щоб наказати їм виконувати самознищення, ймовірно, є одним з найбільш ефективних контрзаходів на рівні інфраструктури.

2.3.2.3 Експлуатація ботнет системи та речей

Стратегії, які базуються на експлуатації використовують той факт, що навіть ботнети містять помилки та дефекти програмування, що призводить до вразливостей, які можуть бути використані для отримання контролю над центральним компонентом (наприклад, C&C-сервером) або через пристрої, заражені ботнетом. Такі уразливості можуть варіюватися від неправильної конфігурації, наприклад, створення незахищеного сервера IRC, яке дозволяє іншим користувачам контролювати канал, до прогавин в безпеці ПЗ, наприклад, перезавантаження буфера, яке можна виконати віддалено.

Стратегії пом'якшення наслідків та маніпулювання не є агресивною для самих заражених машин. Винятком є команди, які завантажують та виконують програми. Експлуатація помилок є ще більш агресивною, ніж виконання регулярних програм, оскільки експлуатаційний код часто потребує спеціальної адаптації до цільової операційної системи та мови. Фреймворки, такі як metasploit, допомагають розробляти загальний експлуатаційний код. Загалом, існує ще більший ризик того, що віддалені системи будуть розбиті таким чином. Це слід враховувати, особливо в сценаріях, де заражені системи контролюють критичні інфраструктури.

Перш ніж використовувати помилки, необхідно знайти заражені системи. Для децентралізованих топологій їх можна перерахувати шляхом підрахунку спроб підключення до введених ботів. У прямих топологіях ця інформація може бути витягнута з даних осідання. Інші варіанти - це використання honeypots, підписів IDS або сканерів, які сканують діапазони мереж, в пошуках заражених пристроїв. У дуже рідкісних випадках списки інших ботів доступні від центральних серверів IRC C & C.

Достовірні вразливості в ботах знаходили і раніше. Багато варіантів Rbot та Sdbot мають однакову кодову базу, яка містить вразливі функції, подібні до цього. Потенційним способом знищення ботнетів буде виявлення заражених машин, використання вразливості в боті та ін'єкцій виконавчого коду, який вимикає шкідливі програми. Вразливий код все ще можна знайти в недавньому шкідливому програмному забезпеченні. Conficker.B використовує MD6 криптографічну хеш-функцію для своїх цифрових підписів. Було встановлено, що алгоритм MD6 містить вразливість буферного розриву та може бути виправлений у випуску оновлення, яке було негайно включено в Conficker.C. Хоча ця специфічна вразливість в Conficker.B не була використана, це показує, що навіть складне шкідливе програмне забезпечення не захищено від критичних прогалин безпеки.

2.3.2.4 Висновок як можуть допомогти проактивні стратегії протидії

Кількість технічно прийнятних стратегій показує, що існує безліч можливостей активно діяти проти ботнетів, перш ніж вони завдадуть шкоди. Хоча це є технічно можливим, на практиці слід враховувати етичні та юридичні проблеми, які виникають у цих стратегіях.

Загальним викликом щодо багатьох наступальних підходів є те, що вони повинні виконуватися максимально приховано. Команди розробників ботнетів можуть протидіяти особливим спробам зменшення наслідків. Можливості маніпуляції можуть бути застарілими при невеликих змінах протоколу або використання цифрових підписів. Крім того, помилки, які використовуються зазвичай можна виправити за короткий час. У випадку, якщо ботнет потрібно вимкнути, це потрібно зробити глобально і швидко, щоб не залишати команді, яка контролює ботнет будь-якого часу для проведення контрзаходів.

Експерти вважають, що переслідування розробників ботнету навряд чи матиме сильний вплив на глобальну загрозу. Натомість з ботнетами потрібно боротися на технічному рівні. Проактивні заходи повинні бути зроблені спільними зусиллями груп міжнародної безпеки з місцевою владою.

2.4 Аналіз особливостей IoT

Щоб правильно вибрати стратегію дій для захисту систем у динамічній та стрімкій сфері IoT – необхідно спочатку проаналізувати пристрої інтернету речей та технологію в цілому. А саме: як вона працює та на яких технологіях базується. Тож в цьому розділі буде розглянуто основні складові систем взаємодії IoT та пристроїв як закінчених ЕОМ, призначених для використання кінцевим користувачем, з яких апаратно-програмних засобів та технологій складається типовий IoT пристрій, як працює криптографічне шифрування у каналах передачі IoT систем та як відбувається оновлення ПЗ. А також для розуміння принципу дослідження IoT зловмисниками буде проаналізовано типовий трафік пристроїв.

2.4.1 Архітектура побудови IoT

Найчастіше архітектура IoT системи складається з 4 етапів. Ці етапи характеризують обробку інформації тими чи іншими пристроями. Етап 1 архітектури IoT складається з мережевих речей, як правило, бездротових датчиків і виконавчих механізмів. Етап 2 включає в себе дані агрегації даних датчиків і перетворення аналого-цифрових даних. На третьому етапі крайові IT-системи виконують попередню обробку даних, перш ніж вони переміщуються до датацентру або хмарного серверу. Нарешті, на 4 етапі дані аналізуються та зберігаються в традиційних системах централізованого центру обробки даних. Очевидно, що стан датчиків / приводів є сферою відповідальності операторів-технологів (OT). Таким чином, це етап 2. Етапи 3 та 4, як правило, контролюються IT, хоча місце обробки обробки може бути на віддаленому сайті або ближче до центру обробки даних. Штрихована вертикальна лінія, позначена як "край", є традиційним розмежуванням обов'язків між OT та IT, хоча це певним чином розмито.

Датчики збирають дані з середовища або об’єкта, який вимірюють і перетворюють його в корисні дані. Приводи можуть також втручатися, щоб змінити фізичні умови, які генерують дані. Привід може, наприклад, вимкнути живлення, регулювати клапан подачі повітря або перемістити роботизований затиск у

процесі зборки.

Рисунок 2.4 – Екосистема IoT

якості повітря, акселерометрів та моніторів серцебиття. І масштаби IoT швидко зростають, частково завдяки технологіям мережевих бездротових сенсорів низької потужності та Power over Ethernet, що дозволяє пристроям у дротовій локальній мережі працювати без потреби в джерелі AC живлення.

У архітектурі IoT деяка обробка даних може відбуватися в кожному з чотирьох етапів. Однак, всі дані, які можна обробляти на датчику – обмежуються потужністю обробки, доступною на кожному з пристроїв IoT. Дані – основа архітектури IoT, і потрібно вибирати між швидкідстю передачі та глибиною обробки цих даних. Чим безпосередніша потреба у інформації, тим ближче до кінцевих пристроїв обробка повинна бути. Щоб глибше зрозуміти, що потребує більш широкої обробки, вам доведеться перемістити дані в хмарне або центр

обробки даних, який може об'єднати кілька джерел даних. Але деякі рішення просто не можуть чекати глибокої обробки.

Дані з датчиків зчитуються в аналоговій формі. Ці дані повинні бути об'єднані та перетворені в цифрові потоки для подальшої обробки. Системи збору даних (DAS) виконують ці функції агрегації та перетворення даних. DAS підключається до мережі датчиків, об'єднує виходи та здійснює аналого-цифрове перетворення. Інтернет-шлюз отримує агреговані та оцифровані дані та маршрути через Wi-Fi, дротові LAN або Інтернет, до етапу 3 систем для подальшої обробки. Системи другого етапу часто знаходяться в безпосередній близькості від датчиків і виконавчих механізмів. Наприклад, насос може містити півдюжини датчиків та виконавчих пристроїв, які передають дані в агрегатор даних, що оцифровує дані. Цей пристрій може бути фізично прикріплений до насосу. Сусідній шлюзовий пристрій або сервер потім оброблятимуть дані та пересилатимуть їх до систем етапу 3 або етапу 4. Попередня обробка даних необхідна, тому що аналогові потоки даних, що походять від датчиків, швидко створюють великі обсяги даних. Вимірювані якості фізичного світу – рух, напруга, вібрація тощо - можуть створювати великі обсяги постійно мінливих даних. Не так багато даних з датчиків може сформувати складна машина, така як двигун повітряного судна за один день, і не існує теоретичної межі кількості датчиків, які можуть передавати дані в систему IoT. Більше того, система IoT завжди включена, забезпечуючи постійне підключення та канали передачі даних. Потік даних IoT може доходити до величезних обсягів, наприклад 40 ТБ / с – реальний потік для системи рівня міської системи вуличного спостереження. Це завелика кількість даних для транспортування в центр обробки даних. Тож краще цю інформацію обробити. Інша причина не передавати дані в ЦОД у цій формі полягає в тому, що аналогові дані мають специфічні терміни та структурні характеристики, які вимагають обробки спеціалізованого програмного забезпечення. Найкраще спочатку перетворити дані в цифрову форму, і це відбувається на другому етапі. Інтелектуальні шлюзи можуть будуватися на додаткових функціях основного шлюзу, додавши такі можливості як: аналітика, за-

хист від шкідливих програм і послуги з управління даними. Ці системи дозволяють аналізувати потоки даних у режимі реального часу. Незважаючи на те, що доставка бізнес-інформації від даних трохи менш безпосередньо на шлюзі, ніж це було б безпосередньо від зони датчика / привода, шлюз має обчислювальні потужності для надання інформації у формі, більш зрозумілій для зацікавлених сторін для бізнесу.

Шлюзи - все ще крайові пристрої – вони є зовнішніми для ЦОД, отже географія та місцеположення – важливі. У прикладі про насос, якщо є 100 насосних агрегатів і необхідно обробляти дані у локальній мережі, можуть бути наявні миттєві дані з насоса, необхідно агрегувати інформацію, щоб створити представлення в цілому по об'єкту, і передавати дані у ЦОД для загального перегляду. DAS і шлюзові пристрої можуть потрапляти в безліч різних середовищ, від підлоги на заводі до мобільних польових станцій, так що ці системи, як правило, спроектовані мобільними, легкими в установці та достатньо міцні, щоб протистояти змінам температури, вологості, пилу, і вібраціям. Після того, як дані IoT були оцифровані та об'єднані, вони готові перейти до сфери IT. Однак дані можуть потребувати подальшої обробки, перш ніж вони потраплять в центр обробки даних. Саме тут починають працювати IT-системи, які проводять більше аналізу. Крайові системи обробки можуть розташовуватися в віддалених офісах або інших місцях розташування крайових пристроїв, але, як правило, вони розташовані на об'єкті або в місці, де датчики розташовані ближче до датчиків, наприклад, у шафі для електропроводки. Оскільки дані IoT можуть легко понизити пропускну здатність мережі та змусити пригнічити ресурси вашого центру обробки даних, тож найкраще мати системи аналітики на краю, які здатні зменшити навантаження на основну IT-інфраструктуру. Також виникають проблеми безпеки, проблеми зберігання та затримки обробки даних. Тож 4-етапний підхід до інфраструктури та обробки IoT потребує нових рівнів співпраці, оскільки розподіл між цими етапами починає розмиватися.

2.5 Перелік проблем, які має вирішувати пропоноване рішення

Вшиті засоби безпеки. Ботнет Mirai продемонстрував, що навіть проста в реалізації атака по словникам може призвести до сотень тисяч інтернет-підключених пристроїв. Хоча рандомізовані паролі за замовчуванням будуть першим кроком, цілком імовірно, що атаки майбутнього розвиватимуться для цільової вразливості програмного забезпечення у пристроях IoT, подібно до ранніх Code Red і Confickr-хробаків. Щоб зменшити цю загрозу, для початку, безпека IoT повинна відійти від відкритих портів за замовчуванням, до закритих за замовчуванням, і прийняти передові практики посилення безпеки. Пристрої повинні приймати до уваги стандартні мережеві конфігурації, які обмежують доступ до віддаленої адреси для цих пристроїв локальним мережам або окремим постачальникам. Окрім забезпечення мережевої безпеки, розробникам IoT необхідно застосувати ASLR, межі розмежування. З точки зору відповідності, сертифікація може допомогти споживачам отримати більш безпечний вибір, а також натиснути на виробників для створення ними більш безпечних продуктів.

Автоматичні оновлення. Автоматичні оновлення, вже стандарт у просторі настільних і мобільних операційних систем, вони надають розробникам механізм своєчасного для виправлення помилок та вразливостей без обтяження споживачів технічними завданнями або вимаганнями відклику. Для автоматичних оновлень потрібна модульна архітектура програмного забезпечення, за допомогою якої можна безпечно перезаписати основні модулі з можливістю відкату у разі виходу з ладу. Вони також вимагають криптографічних примітивів для пристроїв з обмеженим ресурсом та створення інфраструктури PKI для підтримки надійних оновлень. Окрім цих викликів, оновлення також вимагає, щоб спільнота IoT активно відстежувала в себе вразливі місця, що є потенційно обтяжливою відповідальністю з огляду на різноманітність пристроїв. Нагорода за пошук помилок може допомогти у цьому відношенні: приблизно 25% усіх вразливостей, які було виправлено в 2015 році в браузерах Chrome та Firefox, отримано за допомогою цієї системи. А Netgear тільки в 2017

році запустив банер для помилок для ПЗ у своєму маршрутизаторі. У випадку експлуатації з нульовим днем, яка вимикає автоматичні оновлення – розробники IoT повинні забезпечити безпечний механізм резервного копіювання, який, ймовірно, потребує фізичного доступу та втручання споживачів.

Зараження Deutsche Telekom та подальші виправлення дають чудовий приклад цього питання. Маршрутизатори DT мали вразливість, яка дозволила розповсюджувати ботнет за допомогою свого механізму оновлення, що нагадує про те, що основне посилення безпеки має бути першим пріоритетом. Однак, оскільки DT мав автоматичний механізм оновлення, він також міг швидко виправляти пристрої, вимагаючи мінімальне втручання користувача. Впровадження автоматичних оновлень на пристроях IoT не є неможливим, але він повинен розроблятися з більшим усвідомленням проблематики та з більшими зусиллями інженерів.

Сповіщення. Сповіщення через позаполосні канали слугують механізмом резервного копіювання, щоб повернути пристрої у відповідність вимогам безпеки чи видаляти інфекції. Приклади останніх років включають в себе сповіщення адміністраторів пристроїв через посилання CERT, зловживаючи контактом через електронну пошту у записах WHOIS, а також в попередженнях браузерів власникам сайтів про те, що їх сторінка скомпрометована. Сповіщення в просторі IoT з технічної точки зору носять складний характер. IoT пристроям не вистачає як публічної ознаки власності, так і встановленого каналу зв'язку для досягнення споживачів. Якщо споживачі доступні, тоді має бути чіткий і простий шлях оновлення – для вирішення проблеми. Як мінімальна альтернатива, пристроям IoT бути необхідно зареєструвати адресу електронної пошти з виробником або з уніфікованою та сумісною моніторинговою платформою, яка може попереджати споживачів про серйозні проблеми. Це простір, де IoT вимагає нетехнічного втручання. Проблема простоти використання, яка полягає в тому, щоб діяти на сповіщеннях, залишається відкритою проблемою дослідження.

Сприяння ідентифікації пристрою. Щоб покращити метод ідентфікації – виробники IoT можуть прийняти єдиний спосіб ідентифікації версії моделі та прошивки до мережі, скажімо, кодуючи їх у частині MAC-адреси пристрою. Розкриття цієї інформації на рівні 2 зробить його видимим операторам локальної мережі (або домашньому маршрутизатору користувача), яка зможе робити автоматичні дії, щоб відключити віддалений доступ до відомого вразливого апаратного забезпечення до його оновлення. Досягнення цього в єдиній формі у всій галузі, швидше за все, вимагатиме прийняття стандартів.

Дефрагментація. Фрагментація створює ризик безпеки (і сумісності) для підтримки та управління пристроями IoT. Під час сканування було виявлено численні реалізації Telnet, FTP та HTTP стеків. Співтовариство IoT відреагувало на це завдання, прийнявши декілька операційних систем, приклади яких включають Android Thing, RIOT OS, Tock і Windows для IoT [30]. Цей поштовх у бік дефрагментації призведе до абстрагування від нюансів безпеки, які вимагаються від запропонованих жорстких рішень.

Термін актуальності. Навіть з урахуванням найкращих практичних методів роботи, термін служби може залишити сотні тисяч вживаних пристроїв IoT без підтримки. Відсутність довготривалої підтримки призведе до створення двокласової системи захищених та незахищених пристроїв, подібних до поточного стану машин Windows XP. З часом, ризик того, що ці пристрої створять більшість в Інтернеті, будуть зростати. Зупинити цей процес можна буде лише якщо такі пристрої будуть виведені в офлайн-режим.

2.6 Дослідження найбільш вразливих вузлів IoT

За даними дослідження Kaspersky Lab за другий квартал 2018 року (таблиця 2.3), найчастіше атакам піддаються «межеві пристрої», які є транспортерами та агрегаторами даних з датчиків та сенсорів, а саме шлюзи та роутери. Пристрої у даній точці спостереження можуть перевіряти, зберігати, керувати та блокувати будь-який мережевий трафік, що перетинає його шлях. Весь

трафік між пристроями WiFi у локальній мережі або з пристроїв в Інтернет перетинає цю межу.

В аналітичному розділі досліджувалися основні цілі атаки і було виявлено, що основною причиною компрометації та заражень є механізми оновлення (табл. 1.1). А так як вимкнення механізму оновлень або перекриття доступу пристроям IoT до мережі інтернет неможливе, через те, що це знівелює основну цінність технології та поставить їх під удар, тоді залишається лише один спосіб протидії та боротьби з ботнет у системах IoT – це аналіз мережевого трафіку, визначення загроз та блокування такого трафіку.

Тож, мета полягає у виявленні та запобіганні трафіку атак, що походить від пристроїв в межах локальної мережі. Будь-який пристрій, підключений до мережевого пристрою, може надсилати як нормальний трафік, так і трафік атаки протягом одного періоду часу. Кожен пристрій також здатний проводити різні різні атаки послідовно, а послідовні атаки можуть варіюватися за тривалістю. Це відображає, як через канал управління C&C віддаленого ботнету можна змінювати порядки атаки. Припускаємо, що часовий діапазон DoS-атак становить приблизно 1,5 хвилини, загальна тривалість для атак DoS, що намагається уникнути виявлення.

2.7 Алгоритм аналізу аномалій

Алгоритм виявлення аномалій виглядає наступним чином:

1. Збір трафіку. Процес захоплення трафіку записує вихідну IP-адресу, порт джерела, IP-адресу призначення, порт призначення, розмір пакета та часовий покажчик всіх IP пакетів, відправлених з IoT пристроїв.

2. Групування пакетів по Пристрою та Часу. Кожен пристрій IoT розпізнається по власній IP-адресі. Пакети з кожного пристрою далі поділяються на непересічні часові проміжки, записаними межевому пристрої.

3. Витяг параметрів. Stateless та Stateful параметри створюються для кожного пакета, походження якого відоме з поведінки пристрою IoT. Stateless па-

раметри переважно є заголовками пакетів, тоді як stateful параметри - сукупна інформація про потоки за дуже короткі часові проміжки.

2.7.1 Методологія збору трафіку

Була створена експериментальна мережа пристроїв IoT, щоб збирати типовий та зловмисний трафік з пристроїв IoT (Рисунок 2.6). Був налаштований Raspberry Pi v3 як точка доступу Wi-Fi, щоб діяти як посередник. Потім підключена камеру YI та Belkin WeMo Smart Switch до мережі Wi-Fi Raspberry Pi. Монітор артеріального тиску Myings також був підключений Bluetooth до смартфона на ОС Android, пов'язаного з мережею Wi-Fi.

Щоб зібрати звичайний (не-DoS) трафік, відбувалася взаємодія з усіма трьома пристроями IoT протягом 10 хвилин і записувалися pcap-файли, реєструючи всі пакети, відправлені протягом цього періоду часу. Було проведено безліч взаємодій, які трапляються під час звичайного використання пристрою, включаючи потокове відтворення відео з камери YI на сервер у режимах HD та RD, вмикання / вимикання WeMo Smart Switch та встановлення оновлень прошивки, збирання вимірювань артеріального тиску з кров'яного тиску тавідправка вимірювань на сервер для зберігання. Потім було відфільтровано весь трафік, не пов'язаний з IoT, із записів pcap, у тому числі фонового трафіку з телефону Android.

Щоб уникнути ризиків безпеки та складності керування справжнім кодом ботнет, було імітувано три найпоширеніші класи DoS-атак, які запускатимуть Mirai-інфікований пристрій: флуд протоколу TCP, SYN, UDP та HTTP GET flood. Була використана віртуальна машину Kali Linux на ноутбуці, в якості джерела DoS, а Raspberry Pi 2 запускав Apache Web Server як жертву DoS. Було з'єднано обидва пристрої за допомогою Wi-Fi, де точкою доступу виступав Raspberry Pi 3. Тоді джерело DoS націлило атаку на IP-адресу жертви. Було проведено кожен клас DoS-атак приблизно по 1,5 хвилини кожної. Точка доступу фіксувала PCAP файли трафіку атаки, за допомогою інструмента Dumpcap Linux. HTTP GET Flood був змодельований за допомогою інструмента Goldeneye. TCP SYN Flood та UDP Flood були змодельовані за допомогою утиліти hping3 Kali Linux.

Потім трафік DoS був об’єднаний із звичайним трафіком, підробляючи IP-адреси джерела, MAC-адреси та час відправки пакетів, щоб зробити його таким, як ніби пристрої IoT одночасно виробляли звичайний трафік і проводили атаки DoS. Протягом 10 хвилин кожен з трьох IoT-пристроїв, виконав кожен з трьох класів атаки DoS один раз. Атаки виконувались у випадковому порядку та випадкової тривалості, стабільно коливаючись від 90 до 110 секунд кожна. Таким чином, було зібрано приблизно 300 секунд (5 хвилин) трафіку атаки на пристрій. Розподіл атак між пристроями був незалежним.

У результаті цього процесу склав набір даних із 491 855 пакетів, що складаються з 459 655 зловмисних пакетів і 32 290 звичайних.

2.7.2 Аналіз отриманих пакетів та визначення параметрів по пакетам

Досліджується два класа параметрів і аналізується чому вони мають відношення до диференціації нормального трафіку та трафіку атаки IoT.

Stateless параметри можуть бути отримані з незалежних від потоку характеристик з окремих пакетів. Ці параметри створюються без розгрупування вхідного потоку трафіку по IP-адресі. Таким чином, ці особливості є найбільш легкими. Stateless параметри:

1. Розмір пакету. Розподіл розмірів пакетів значно відрізняється від атаки та звичайного трафіку (рисунок 2.7). Більше 90% пакетів атаки становлять менше 100 байт, а звичайні - від 100 до 1200 байт. Пристрій, що проводить атаку DoS, такий як TCP SYN Flood, намагається відкрити якомога більше запитів на з'єднання з жертвою, щоб вичерпати ресурси сервера жертви. Таким чином, зловмисник хоче зберегти розмір пакетів якомога менше, щоб максимально збільшити кількість запитів на підключення в секунду. Для порівняння, звичайний трафік може варіюватися від простого пінгування серверів, які вказують на те, що пристрій активний (невеликі пакети) до даних потокового відео (великі пакети).

2. Міжпакетний інтервал. Нормальний трафік IoT має обмежене значення скупчення (Рисунок 2.7). Більшість пакетів надсилаються регулярними інтервалами з помітним часом між пакетами. Це може відображати мережеві зв'язки IoT або інші автоматичні дії мережі. На відміну від цього, переважна більшість трафіку атаки DoS має близькі до нуля міжпакетні інтервали (ΔT) і високі перші та другі похідні міжпакетних інтервалів.

3. Протокол. Звичайний та трафік атаки також мають різні розподіли протоколів (рисунок 2.8). Трафік атаки включає менше протоколів аніж звичайний трафік та містить набір, нерелевантних до процесу роботи пристрою, протоколів.

Stateful параметри показують, як мережевий трафік розвивається з часом. У процесі генерації цих параметрів є накладні витрати, оскільки необхідно розбити мережевий трафік на потоки для кожного пристрою та розкладаємо потоки кожного пристрою у кожен проміжок часу. Stateful параметри:

1. Пропускна здатність. Алгоритм розгруповує мережевий трафік за джерелом пристрою і обчислює середню пропускну здатність протягом 10 секундних часових проміжках, щоб виміряти середню пропускну здатність кожного пристрою. Існують незначні відмінності у пропускній здатності між звичайним та атакованим трафіком (Рисунок 2.8). Передбачається, що модель машинного навчання (ML) зможе використати ці відмінності.

2. IP-адреса призначення. IoT пристрої характеризуються обмеженою кількістю кінцевих точок, з якими вони спілкуються. Наприклад, комутатор WeMo спілкується лише з чотирма кінцевими точками для активації / деактивації з хмарного серверу та отримання оновлень ПЗ (Рисунок 2.8). Іншою ключовою характеристикою трафіку пристрій IoT є те, що встановлена IP-адреса призначення рідко змінюється з часом.

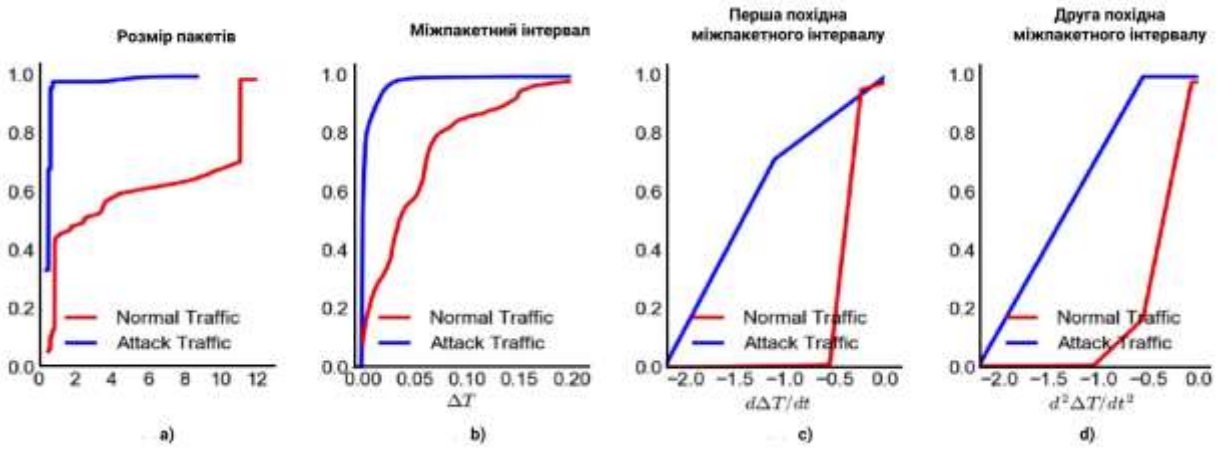


Рисунок 2.7 – Порівняння статистики параметрів трафіку атаки та звичайного трафіку. a) Розміри пакетів (b-d) Інтервали між пакетами ΔT, dΔT / dt та dΔT / dt2

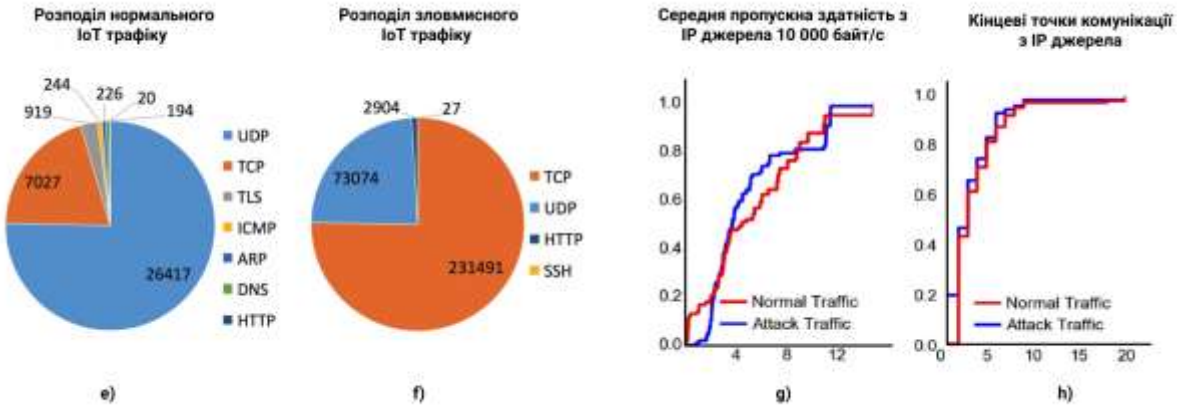


Рисунок 2.8 – e-f – розподіл протоколів. g – середня пропускна здатність протягом більше 10 секунд. h – кількість унікальних IP-адрес у проміжку 10 секунд

2.8 Висновок до другого розділу

У роботі був проводений аналіз ботнет, його методів реалізації та досліджені особливості систем IoT. Отримані дані свідчать про те, що найбільш ефективним та доцільним методом протидії ботнет та виявлення аномалій у системах IoT – це аналіз мережевого трафіку для виявлення не типової аномальної активності системи.

Був проведений аналіз, який довів ефективність такого методу.

РОЗДІЛ 3. ЕКОНОМІЧНИЙ РОЗДІЛ

В даному розділі проводяться економічні розрахунки витрат на дослідження та аналіз типового та зловмисного мережевого трафіку пристроїв IoT з використаннням моделей, розробницьких стендів та образчиків пристроїв IoT.

3.1 Особливості розробки алгоритму аналізу та моделювання

Розробка і реалізація моделі є складним процесом, який має специфічні особливості. Його розробка, створення і апробація здійснюються за фазами життєвого циклу. Він включає три стадії: розробка (проектування), моделювання (створення) і використання (оцінка ефективності розробленої експериментальної моделі). Кожна стадія поділяється на етапи:

- проектування;
- реалізація;
- тестування і випробування алгоритму аналізу;
- аналіз результатів тестування.

Виходячи з часу виконання дипломного проектування, був розроблений графік виконання завдання. Діаграма розподілу часу виконання робіт представлена на рисунку 3.1.

До етапу «проектування» відноситься аналіз технічного завдання на розробку, але, перш за все, аналіз та дослідження можливостей збору та аналізу мережевого трафіка пристроїв IoT. З пояснювальної записки сюди потрібно віднести написання вступу і першого розділу.

До етапу «реалізація» відносяться огляд можливих варіантів реалізації такого аналізу, створення моделі та налаштування ПЗ для безпосереднього збору та обробки інформації. З пояснювальної записки на цьому етапі проводиться написання другого розділу.

На етапі «тестування» проводиться тестування алгоритму аналізу та моделі з різними параметрами, включаючи неможливі.

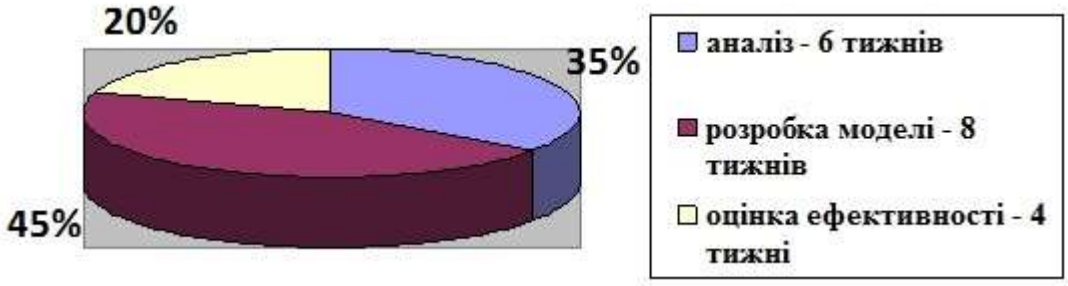


Рисунок 3.1 – Діаграма розподілу часу виконання робіт

3.2 Визначення трудомісткості розробки алгоритму збору та аналізу мережевого трафіка пристроїв IoT.

Трудомісткість продукції – показник, який характеризує витрати робочого часу на виробництво будь-якої споживчої вартості або на виконання конкретної технологічної операції.

Трудомісткість розробки алгоритму можливо розрахувати по формулі:

$$t = t_O + t_A + t_Л + t_P + t_Д \text{ , год.,}$$

(3.1)

де t_O – витрати праці на підготовку і опис поставленого завдання;

t_A – аналіз та дослідження машинного навчання;

$t_Л$ – тривалість вивчення літературних джерел за темою тощо;

t_P – витрати праці на розробку та написання програми;

t_m – витрати на тестування програмного забезпечення;

$t_Д$ – витрати праці на оформленням документації (за умови роботи однієї людини).

Оцінка витрат праці на кожен показник залежить від конкретних умов і визначається тривалістю окремого робочого процесу (табл. 3.1). Зважаючи на той факт, що дослідження, пов'язані з мережевими пристроями IoT:

Таблиця 3.1 – Тривалість робочих процесів

Назва робочого процесу	Тривалість, год.
Підготовка і опис поставленого завдання	$t_O = 4$
Вивчення літературних джерел	$t_Л = 90$
Проектування та реалізація моделі	$t_P = 20$
Збір мережевого трафіка	$t_A = 30$
Обробка результатів	$t_m = 3$
Підготовка документації по завданню	$t_Д = 15$

Таким чином, визначивши трудомісткість окремих показників, розрахуємо сумарну трудомісткість розробки експериментальної моделі по формулі 3.1:

$t = 4 + 30 + 90 + 20 + 3 + 15 = 162$ *год*

3.3 Розрахунок витрат на розробку алгоритму

Витрати на розробку програмного забезпечення $K_{ЕК}$ включають витрати на заробітну плату інженерів кібербезпеки $З_{ЗП}$ і вартість машинного часу $З_{МЧ}$, необхідного для розробки моделі, і розраховуються за формулою:

$$K_{ІМ} = З_{ЗП} + З_{МЧ}\ , грн. \tag{3.2}$$

Заробітна плата – винагорода, обчислена, у грошовому виразі, яку за трудовим договором власник або уповноважений ним орган виплачує працівникові за виконану ним роботу. Розмір зарплати залежить від складності та умов виконуваної роботи, професійно-ділових якостей працівника, результатів його праці та господарської діяльності підприємства.

Заробітна плата виконавців визначається за формулою:

$$З_{ЗП} = t \cdot С_{ЗП}, грн. \quad (3.3)$$

де t – загальна трудомісткість розробки моделі, яка визначається за формулою 3.1, годин;

$С_{ЗП}$ – середньогодинна заробітна плата інженера в галузі телекомунікацій, (основна і додаткова), з урахуванням відрахувань на соціальні потреби, грн/годину.

Визначення мінімальної середньої годинної оплати інженера кібербезпеки обчислюється з урахуванням 8-ми годинного робочого графіку на добу і 5-ти денної робочої неділі, та знаючи його середній щомісячний оклад.

На 01.11.2018 року оклад складає 8000 грн. Єдиний соціальний внесок складає 22%, тобто 1760 грн. Отже, з урахуванням премій (20%), можливих надбавок (10%) і відрахувань на соціальні потреби, заробітна плата інженера кібербезпеки складає 12160 грн.

Таким чином, середня заробітна плата за одну годину роботи становить:

$$С_{ЗП} = \frac{12160}{176}$$

Таким чином, витрати на оплату праці розробника, з урахуванням форму-

$$З_{ЗП} = 162 \cdot 69$$

ли 3.3, складають:

Розрахунок вартості машинного часу, необхідного для розробки алгоритму та імітаційної моделі включає витрати на необхідне програмне та апаратне забезпечення і витрати на електроенергію, і здійснюється за формулою 3.4:

$$З_{МЧ} = С_О + С_{ЕЛ}, грн. \quad (3.4)$$

де $С_О$ – витрати на обладнання. Відповідні дані наведені в таблиці 3.2;

$С_{ЕЛ}$ – витрати на електроенергію, грн.

Так як операційна система (дистрибутив) Kali Linux має вбудований модуль вивчення мережевого трафіка та роботи з ним, то існують лише затрати на обладнання.

Таблиця 3.2 – Вартість необхідного апаратного забезпечення

Найменування	Вартість, грн
Ноутбук HP Pavilion g5-1311SE	3500,00
Миша Assero 68007-BK	178,00
Raspberry Pi 3 2 шт.	2000,00
WeMo Smart Switch	370,00
Камера YI	630,00
Разом:	6678,00

Витрати на електроенергію ($С_{ЕЛ}$) залежать від часу роботи на ЕОМ ($Т_{ЕОМ}$) та собівартості машино-години роботи ЕОМ ($С_{МЧ}$), і розраховується за формулою:

$$С_{ЕЛ} = С_{МЧ} \cdot Т_{ЕОМ}, грн. \quad (3.5)$$

Розрахунок вартості машино-години ЕОМ проведемо по формулі 3.6:

$$C_{МЧ} = W \cdot Ц_{ЕЛ}, грн/год.,$$

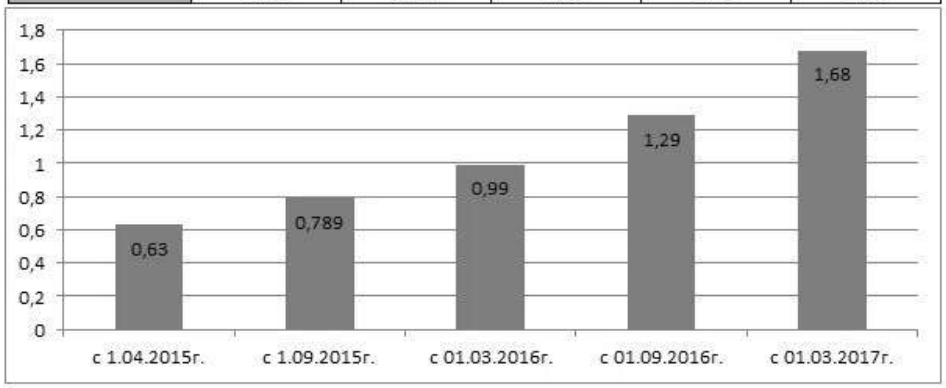
(3.6)

Де W – потужність ЕОМ, *W=0,5 кВт/год;*

$Ц_{ЕЛ}$ – вартість 1кВт за годину електроенергії.

Поточні тарифи на електроенергію для населення України, встановлені відповідно до постанови НКРЕКУ України № 220 від 01.03.2017 року (діючі з 1

	с 01.04.2015	с 01.09.2015	с 01.03.2016	с 01.09.2016	с 01.03.2017
За 1 кВт*час	0,63	0,789	0,99	1,29	1,68



березня 2017 року) представлені на рисунку 3.2.

Рисунок 3.2 – Тарифи на електричну енергію, для технічних цілей, яка витрачається в багатоквартирних будинках та гуртожитках.

$$C_{МЧ} = 0.5 \cdot 1.68 = 0.84, грн/год.$$

Таким чином, вартість машино-години ЕОМ за формулою 3.6 складе:

Час роботи на ЕОМ складає фактичні витрати часу на розробку алгоритму та імітаційної моделі. Згідно з рисунком 3.1, тривалість зайняла 18 тижнів

або 126 днів. З урахуванням того, що ЕОМ працювала в середньому по 5 годин

$$T_{ЕОМ} = 126 \cdot 5 = 630 \; год.$$

на добу отримуємо:

$$C_{ЕЛ} = 0.84 \cdot 630 = 529, \; грн.$$

Таким чином вартість електроенергії, за формулою 3.5, складатиме:

Враховуючи відому вартість витрат на обладнання та витрачену електроенергію проведемо розрахунок вартості машинного часу, який є необхідним для

$$З_{МЧ} = 6678 + 529$$

розробки алгоритму та імітаційної моделі на ЕОМ за формулою 3.4:

Отже, витрати на розробку моделі в середовищі Matlab Simulink склада-

$$К_{ПЗ} = 7207 + 11178$$

ють, виходячи з формули 3.2:

Визначені таким чином витрати на розробку моделі та алгоритму збору та аналізу мережевого трафіка пристроїв IoT, є одноразовими капітальними витратами і складають 18385 грн.

Також до затрат треба віднести витрати на «матеріали», які враховують: витрати на носії даних, папір для друкувальних пристроїв і архівну обробку документації. В процесі розробки було потрібно:

- CD-диск вартістю 8 грн.;
- Аркуші паперу формату А4 для друку документації, вартість одного аркуша 0,6 грн. Загальна сума склала 55 грн.;
- Обкладинка документів, вартість за роботу 75 грн.

$$З_м = 8 + 55 + 75 = 138, грн.$$

Загальна вартість витрат на матеріали складає:

Діаграма відображає вагу обчислених значень вартості розробки експериментальної (рис. 3.3)

Рисунок 3.3 – Склад капітальних витрат на розробку моделі

3.4 Розрахунок поточних (експлуатаційних) витрат

Поточні витрати розрахуємо на рік. До них входять витрати на електроенергію та амортизаційні відрахування від вартості обладнання.

Вартість електроенергії, що споживається комп'ютером для функціонування системи протягом року $С_е$ визначається за формулою (3.7):

(3.7)

$$С_{ел} = Р \cdot F_р \cdot Ц_е ,$$

де Р – встановлена потужність апаратури інформаційної безпеки, $F_р$ – річний фонд робочого часу системи інформаційної безпеки (визначається виходячи з режиму роботи системи інформаційної безпеки), $Ц_е$ – тариф на електроенергію, грн/кВт·годин.

Таким чином, вартість спожитої комп'ютером електроенергії за формулою (3.7) складає:

$$С_{ел} = 0.5 \cdot 500 \cdot 1.68 = 420, грн/рік$$

Вартість амортизаційних відрахувань, які відраховані лінійним способом, розраховується за формулою (3.8):

$$C_а = (L_М + C_{ПК}) / t_{експ}, \textit{ грн/рік}$$

де $t_{експ}$ – термін корисного використання програмного-апаратного комплексу,

$C_{ПК}$ – вартість комп'ютера.

$$C_а = 6500 / 5 = 1300, \textit{ грн/рік}$$

3.5 Оцінка можливого збитку від компрометації системи IoT пристроїв

Припустимо що існує мережа пристроїв IoT, компрометація якої спричиняє додавання її до ботнету. А середнє число атак приймемо N = 400. Збитки від компрометації пристроїв складають U = 1 тис грн.

Таким чином загальний збиток від атаки на ці пристрої за допомогою ботнет складає:

$$B = \sum_{N} U$$

$$B = U \cdot 400 = 1000 \cdot 400 = 400000 \text{ грн}$$

Очікувана імовірність атаки на вузол або пристрій корпоративної мережі R приймемо $R = 0.48$

Загальний ефект від впровадження системи захисту від змодельованого каналу з урахуванням щорічних експлуатаційних витрат С складає:

$$E = B \cdot R - C$$

$$E = 400000 \cdot 0.48 - 7.1 = 207992{,}9, \text{ грн}$$

3.6 Визначення показника економічної ефективності

Коефіцієнт повернення інвестицій ROSI

$$ROSI = \frac{E}{K} = \frac{207992{,}9}{18276}$$

Організація здійснює фінансування капітальних інвестицій за рахунок реінвестування власних коштів, тому в якості бажаного значеннння показника ефективності $E_н$ приймаєтьсябажана норма прибутковості альернативних

$$E_Н = (N_{деп} - N_{інф})$$

варіантів вкладення коштів з урахуванням інфляції.

$N_{деп}$ – річна депозитна ставка або прибутковість альтернативного варіанту вкладення коштів, %

$$E_н = \frac{(16\% - 13{,}5\%)}{100}$$

$N_{інф}$ – ріний рівень інфляції

Виходячи з нерівності

$$ROSI > E_н$$

$$11.38 > 0.025$$

можна зробити висновок, що проект є економічно доцільним.

3.7 Розрахунок терміну окупності

$$T_O = \frac{K}{E} = \frac{1}{ROSI}$$

Розрахунок проводиться за формулою:

$$T_0 = \frac{1}{11.38}$$

3.8 Висновки до економічного розділу

В економічному розділі розраховано час, що необхідний для розробки методології збору та аналізу звичайного та зловмисного мережевого трафіку

пристроїв IoT, заробітну плату робітників інженерів кібербезпеки, затрати на матеріали. Встановлено, що затрати на розробку алгоритму, реалізацію і тестування, становлять 38 467 грн., експлуатаційні витрати складають 4 360 грн/рік.

ВИСНОВКИ

У цій роботі було використано декілька розроблених засобів вимірювання активності ботнету, які обґрунтовують занепокоєння щодо безпеки IoT. Також використовувались раніше представлені рішення, як керівництво для власних пропозицій щодо боротьби з ботнет IoT у цілому.

Згідно отриманих результатів проаналізованого мережевого трафіку пристроїв IoT можна зробити висновок, що запропонований алгоритм аналізу параметрів трафіку є ефективним методом порівняння аномального та типового трафіку. Що, надалі, дозволяє робити висновок щодо статусу системи та наявності в ній аномалій.

Для подальшого виявлення загрози, на основі отриманих даних необхідно їх класифікувати та розробити рішення, згідно кожної можливої ситуації. Така обробка може бути виконана, наприклад, за допомогою нейронних мереж або інших методів самонавчання.

Таким чином, результат роботи може слугувати основою для розробки системи виявлення вторгнень у систему IoT, що забезпечує захист системи від ботнет. Для реалізації такої системи необхідно протестувати запропонований метод на більшій кількості пристроїв та зібрати реальний трафік для підтвердження ефективності запропонованого рішення.

ПЕРЕЛІК ПОСИЛАНЬ

1. К. О. Кіфорчук, М. В. Грайворонський – «Оцінка вразливості пристроїв «інтернету речей»»

2. Check Point Software Tech. LTD « Most Wanted Malware: Attacks Targeting IoT and Networking doubled since May 2018» URL: https://blog.checkpoint.com/2018/08/15/julys-most-wanted-malware-attacks-targeting-iot-and-networking-doubled-since-may-2018/

3. Menachem Domb – «An Adaptive Lightweight Security Framework Suited for IoT» URL: https://www.intechopen.com/books/internet-of-things-technology-applications-and-standardization/an-adaptive-lightweight-security-framework-suited-for-iot

4. Felix LEDER, Tillmann WERNER, and Peter MARTINI Institute of Computer Science IV, University of Bonn, Germany – «Proactive Botnet Countermeasures – An Offensive Approaches» URL: http://four.cs.uni-bonn.de/fileadmin/user_upload/leder/proactivebotnetcountermeasures.pdf

5. Ivo van der Elzen Jeroen van Heugten – «Techniques for detecting compromised IoT devices» URL: http://www.delaat.net/rp/2016-2017/p59/report.pdf

6. Kaspersky Lab, Secure List – «DDoS-атаки в третьем квартале 2018 года» URL: https://securelist.ru/ddos-report-in-q3-2018/92512/

7. Kaspersky Lab, Михаил Кузин, Ярослав Шмелев, Владимир Кусков – «Новые тренды в мире IoT-угроз» URL: https://securelist.ru/new-trends-in-the-world-of-iot-threats/91601/

8. Manos Antonakakis – «Understanding the Mirai Botnet»

9. Rohan Doshi, Noah Apthorpe, Nick Feamster – «Machine Learning DDoS Detection for Consumer Internet of Things Devices»

10. Sebastian-Dan Naste – «A multidisciplinary study on DDoS attacks in the EU IoT ecosystem»

11. xaker.ru – «IoT-ботнеты Mirai и Gafgyt» URL: https://xakep.ru/2018/09/11/new-mirai-and-gafgyt/

12. OWASP – «IoT Vulnerabilities Project» URL: https://www.owasp.org/index.php/OWASP_Internet_of_Things_Project#tab=IoT_Vulnerabilities

13.OWASP – «IoT Attack Surface Project» URL: https://www.owasp.org/index.php/OWASP_Internet_of_Things_Project#tab=IoT_Attack_Surface_Areas

14. Daniel Elizalde – «IoT Hardware – Introduction and Explanation» URL: https://www.iotforall.com/iot-hardware-introduction-explanation/

15. Earlence Fernandes та співавтори «FlowFence: Practical Data Protection for Emerging IoT Application Frameworks» URL: https://www.usenix.org/system/files/conference/usenixsecurity16/sec16_paper_fernandes.pdf

16. HESSELDAHL A. «The Hacker's Eye View of the Internet of Things.» URL: http://recode.net/2015/04/07/a- hackers- eye- view- of- the- internet- of- things/.

17. FERNANDES, E., JUNG, J., AND PRAKASH, A.. – «Security analysis of emerging smart home applications». На IEEE Symposium on Security and Privacy (S&P)

18. Yi home camera. URL: https://www. yitechnology.com/yi- home- camera

19. Hewlett Packard Enterprise – «Internet of things research study». URL: http://h20195.www2.hpe.com/V4/getpdf.aspx/4aa5- 4759enw

20.«Internet of things (iot) security and privacy recommendations.».

21. S. Hilton – «Dyn analysis summary of friday october 21 attack.» URL: https://dyn.com/blog/ dyn- analysis- summary- of- friday- october- 21- attack/

22. V.Chandola, A.Banerjee, V.Kumar – «Anomaly detection: A survey» vol. 41.3

23.E. Eskin, W. Lee, and W. Stolfo – «Modeling system call for intrusion detection using dynamic window sizes»

24. M. Qin and K. Hwang – «Frequent episode rules for internet anomaly detection»

25. M. Miettinen, S. Marchal, I. Hafeez, N. Asokan, A. Sadeghi, та S. Tarkoma – «Iot sentinel: Automated device-type identification for security enforcement in IoT»

ДОДАТОК А. Відомість матеріалів дипломного проекту

№	Формат	Найменування	Кількість листів	Примітка
1	А4	Реферат	3	
2	А4	Список умовних скорочень	1	
3	А4	Зміст	2	
4	А4	Вступ	1	
5	А4	1 Розділ	27	
6	А4	2 Розділ	32	
7	А4	3 Розділ	10	
8	А4	Висновки	1	
9	А4	Перелік посилань	2	
10	А4	Додаток А	1	
11	А4	Додаток Б	1	
12	А4	Додаток В	2	
13	А4	Додаток Г	1	

ДОДАТОК Б. Перелік файлів на елетронному носії

1. Магістерська робота Горошко_ЄО_125м-17-1.docx – Пояснювальна записка
2. Горошко_ЄО.pptx – Презентація

ДОДАТОК В. ВІДГУК НА ДИПЛОМНУ РОБОТУ МАГІСТРА

НА ТЕМУ:

«Способи протидії бот-мережам в системах Інтернет речей»

студента групи 125м–17–1 Горошко Євгена Олександровича

Мета дипломної роботи – підвищення рівня захисту систем Інтернет речей від загрози створення на їх базі бот-мереж .

Тема дипломної роботи безпосередньо пов’язана з об’єктом діяльності фахівця за спеціальністю 125 Кібербезпека – розвиток способів захисту інформаційно-телекомунікаційних систем.

Задачі дипломної роботи (аналіз особливостей функціонування систем Інтернет речей, аналіз існуючих вразливостей та загроз для Інтернет речей, аналіз методів та засобів захисту, обґрунтування вибору параметрів для систем виявлення вторгнень) віднесені в освітньо-кваліфікаційній характеристиці магістра до класу евристичних, вирішення яких основане на знаково-розумових уміннях фахівця.

Практичне значення результатів проектування полягає у обґрунтуванні параметрів для систем виявлення вторгнень в системах Інтернет речей.

До недоліків дипломної роботи відносяться:

– відхилення від графіка роботи;

–викладення результатів роботи недостатньо структуровано;

–недостатньо обґрунтовано вибір методів протидії бот-мережам.

Оформлення пояснювальної записки до дипломного проекту виконано з деякими відхиленнями від стандартів.

Ступінь самостійності виконання дипломної роботи висока.

За час дипломування Горошко Є.О. виявив себе фахівцем, здатним самостійно, на достатньо високому рівні вирішувати поставлені задачі.

В цілому дипломна робота виконана у відповідності до вимог, що пред’являються до дипломної роботи магістра, заслуговує оцінки “добре”, а

Горошко Є.О. присвоєння йому кваліфікації професіонала із організації інформаційної безпеки.

Керівник спеціальної частини
дипломної роботи магістра,
старший викладач _____ О.В. Кручінін

Керівник дипломної
роботи магістра,
д.т.н, проф. _____ В.І. Корнієнко

ДОДАТОК Г. Відгук керівника економічного розділу

Керівник розділу _____ _____

(підпис) (ініціали, прізвище)