КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№2 (6), 2019
ISSN 2663 - 4023
DOI 10.28925/2663-4023.2019.6.7181
УДК 004.05

Барибін Олексій Ігорович
кандидат технічних наук, в.о. декана фізико-технічного факультету
Донецький національний університет імені Василя Стуса, Вінниця, Україна
ORCID ID 0000-0002-0897-4454
o.barybin@donnu.edu.ua

Зайцева Еліна Євгеніївна
кандидат технічних наук, доцент кафедри комп'ютерних технологій
Донецький національний університет імені Василя Стуса, Вінниця, Україна
ORCID ID 0000-0002-2135-8146
zaytseva.elina@gmail.com

Бражний Володимир Володимирович
студент 1 курсу СО «Магістр» спеціальності 105 Прикладна фізика та наноматеріали (освітня програма «Технології інтернету речей»)
Донецький національний університет імені Василя Стуса, Вінниця, Україна
ORCID ID 0000-0003-3327-9709
brazhnyi.v@donnu.edu.ua

ТЕСТУВАННЯ БЕЗПЕКИ ПРИСТРОЇВ ІНТРЕНЕТУ РЕЧЕЙ НА БАЗІ МІКРОКОНТРОЛЕРА ESP32

Анотація. В статті на основі аналізу сучасних тенденцій в області IoT технологій визначено сегмент, для якого забезпечення інформаційної безпеки може зіткнутися з недостатністю її рівня. Це пристрої IoT на основі мікроконтролера ESP32, які розроблені і впроваджені в домашніх умовах непрофесіоналами. Запропонована і реалізована на лабораторному масштабі фізична модель непрофесіональної саморобної системи IoT, яка включає в себе пристрій для вимірювання температури на базі ESP32 (малогабаритний ESP32 на основі набору ESP32 DevKit V2 виробництва Espressif та цифровий датчик температури DS18B20), домашню мережу WiFi (на базі маршрутизатора TL-WR841N) і веб-інтерфейс (на базі модуля node-red-dashboard). Початкові умови експерименту включали в себе наступне: використання протоколу UDP, аутентифікація і шифрування передачі даних на основі специфікації WPA2-PSK, рівень кваліфікації зловмисника достатній для використання інструментів Aircrack-ng, Airmon-ng, Airodump-ng, Aireplay-ng, Besside-ng, Wireshark. В результаті експерименту на основі цієї моделі був успішно реалізований сценарій отримання несанкціонованого доступу до переданих даних. Сценарій атаки складається з чотирьох етапів: 1 – отримання несанкціонованого доступу до мережі (зміна режиму мережевої карти на режим моніторингу (Airmon-ng), перегляд доступних точок доступу (Airodump-ng), перехоплення рукостискання, вгадування пароля (Besside-ng); 2 – перехоплення і аналізу мережевого трафіку (Wireshark); 3 – створення підробленого клієнта ESP32 використовуючи отримані раніше дані (Arduino) і підключення його до мережі; 4 – від'єднання оригінального пристрою на базі ESP32 від сервера (Aircrack-ng). Показано, що зловмисник, який має базові знання та навички в роботі із розповсюдженими засобами злому бездротових мереж і базові знання та навички програмування ESP32 може отримати доступ до системи і відправити підроблену інформацію на веб-інтерфейс. Для того, щоб зменшити ймовірність пропонованого сценарію рекомендується використовувати протокол TCP, який на відміну від UDP забезпечує цілісність даних і повідомлення відправника про результати передачі.

Ключові слова: інтернет речей; сценарій атаки; ESP32; WiFi.

ISSN 2663 - 4023

1. ВСТУП

Постановка проблеми та аналіз останніх досліджень і публікацій. Технології інтернету речей (Internet of Things – IoT) стають все більш популярним в сучасному високо технологічному суспільстві. Навіть при тому, що точне визначення IoT є майже недосяжним, ми можемо говорити про IoT, як про підключені до мережі Інтернет об'єкти, які можуть збирати дані, обмінюватися зібраними даними один з одним і надавати оброблені дані користувачу [1]. Завдяки поєднанню великої кількості технологій в IoT не існує структурованого та системного підходу для забезпечення належної інформаційної безпеки в цілому, але тільки для конкретних областей і рішень:

- автономні транспортні засоби, вбудовані системи [2],
- поєднані транспортні засоби, системи охорони здоров'я, Smart Grid [3],
- міжплатформне програмне забезпечення, платформи Smart City, вбудовані пристрої [4],
- міжплатформне програмне забезпечення, IIoT WAN, смарт підприємства [5],
- Sybil атаки в автомобільних мережах, інтелектуальні системи будинку, розумні будівлі [6],
- ризик і контроль IoT [7],
- IoT на підприємстві [8].

У більшості випадків IoT системи, які виробляються професійно, мають певний рівень інформаційної безпеки. Сьогодні, однак, набори пристроїв IoT, які розроблені для проектування і складання на дому стають все більш поширеним явищем. Проекти, які базуються на мікроконтролері ESP32 описані в літературі [9, 10], а також на великій кількості веб-сайтів отримують все більшу і більшу популярність. При реалізації такого проекту IoT для передачі даних користувачу використовується домашня бездротова мережа Wi-Fi. У такій системі загрози для інформаційної безпеки в першу чергу будуть пов'язані з використанням відомих вразливостей бездротових мереж. Відповідно з'ясування сценарію атаки з метою компрометації даних, що передаються від пристроїв на основі ESP32 є актуальною проблемою.

Мета статті. Таким чином, метою даного дослідження є вивчення можливості використання вразливостей бездротових мереж Wi-Fi для компрометації даних, що передаються від пристроїв, які базуються на ESP32. Відповідно до поставленої мети завданнями дослідження є:

- побудувати фізичну модель непрофесійної саморобної системи IoT на базі мікроконтролера ESP32;
- визначити початкові умови експерименту;
- визначити можливий сценарій атаки компрометації даних, що передаються через Wi-Fi мережу з пристрою на основі ESP32.

2. ФІЗИЧНА МОДЕЛЬ

Мікроконтролер ESP32 був випущений на ринок у 2015 році і є видатним пристроєм не тільки через свою низьку ціну. Поєднання в одній мікросхемі Wi-Fi і Bluetooth, двоядерного процесора і багатого набору периферійних пристроїв робить ESP32 лідером в своєму сегменті.

В роботі [11] було продемонстровано, що ESP32 може бути успішно використаний для збору даних і управління різними пристроями через бездротову

мережу набагато краще, ніж його попередник ESP8266. Докладний порівняльний аналіз ESP32 та ESP8266 можна знайти в [11], а автори рекомендують використовувати мікроконтролер для непрофесійних розробників. Як найпростіший приклад використання ESP32 можна навести наступний: користувач з мобільним телефоном може віддалено перевірити стан свого власного розумного будинку (температура, вологість тощо) безпосередньо на пристрої.

Розглянемо основні передумови для побудови фізичної моделі системи, яка може бути використана для забезпечення експерименту з компрометації даних, що передаються з пристрою на базі ESP32.

Аналіз проектів, описаних в [9, 10] і представлений на таких сайтах як [12, 13] та інших показав, що одним з найбільш популярних проектів пристроїв IoT на базі ESP32 є метеорологічної станції. Для реалізації подібного проекту відповідно до класичної архітектури IoT (складається з трьох шарів – пристрої вимірювання, мережа та інтерфейс користувача) вважається необхідним:

• підключити датчики для вимірювання параметрів навколишнього середовища,
• налаштувати домашню мережу Wi-Fi для передачі даних,
• реалізувати простий веб-інтерфейс для прийому даних від датчиків.

В якості основи пристрою вимірювання був використаний набір ESP32 DevKit V2. Для моделювання достатньо підключити тільки датчик температури або комбінований датчик температури і вологості. Виробники пропонують велику кількість таких датчиків з приблизно аналогічними характеристиками. В якості одного з варіантів можна використовувати цифровий датчик температури DS18B20 від Dallas Semiconductor. Характеристики такого датчика можна знайти, наприклад, в [14].

З огляду на той факт, що у фізичній моделі передбачається реалізація домашньої мережі Wi-Fi пропонується використовувати в якості точки доступу один з найпопулярніших в цьому сегменті маршрутизаторів TL-WR841N виробника TP-Link.

В результаті було використано наступне обладнання на рівні датчиків і мережі:

- малогабаритний ESP32 на основі набору ESP32 DevKit V2 виробництва Espressif,
- цифровий датчик температури DS18B20,
- маршрутизатор TL-WR841N.

На рівні інтерфейсу для відображення даних, отриманих від датчика використовується інструмент node-red і, зокрема, модуль node-red-dashboard (рис. 1).

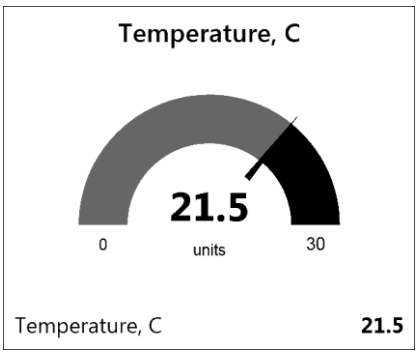

Рис. 1. Відображення значення температури модулем node-red-dashboard

ISSN 2663 - 4023

3. ВИЗНАЧЕННЯ ПОЧАТКОВИХ УМОВ ЕКСПЕРИМЕНТУ

Як уже згадувалося вище, саме на мережевому рівні найвищі шанси на реалізацію атаки на пристрій, основним елементом якої є ESP32. В якості прикладів атак на мережевомутакому рівні можна відзначити Malicious Code Injection, Sniffing Attack, Spear-Phishing Attack, DoS (Denial-of-Service Attack), Sybil Attack, Proxy Attack, Sleep Deprivation attack, і т.д. [1] ,

При передачі даних з пристроїв IoT можуть бути використані різні протоколи, але, як показано в роботах [15, 16] всі вони засновані на TCP або UDP. У нашому випадку ми використовуємо протокол UDP, тому що він базується на простій модель даних і найбільш часто використовується для систем з датчиками через зменшений рівень накладних витрат комунікації. Негативним властивостями цього протоколу є те, що він не гарантує доставку або правильну послідовність дейтаграм і не забезпечує послуг з'єднання і повторної відправки в разі втрати даних.

Механізми захисту мереж Wi-Fi включають в себе перевірку автентичності (клієнт і точка доступу представляються один одному і підтверджують право на обмін даними) і шифрування (вибір алгоритму генерації та ключів шифрування даних). Як зазначено в [17] в домашній WiFi мережі рекомендується використовувати аутентифікацію і шифрування передачі даних на основі специфікації WPA2-PSK.

Вразливості протоколів безпеки Wi-Fi мереж описані, наприклад, в [18]. WPA і WPA2 протоколи шифрують дані окремо для кожного клієнта з тимчасовим ключем, який генерується після того, як клієнт з'єднатися з точкою доступу. Щоб отримати ключ, потрібно знати параметри мережі, які можуть бути вільно перехоплені шляхом підслуховування мережевого трафіку, і головною перешкодою для зловмисника є отримання Pre-Shared Key через перевірку всіх можливих комбінації паролів або з використанням словника. Для підбору пароля зловмисник повинен перехопити рукостискання між клієнтом і точкою доступу. Швидкість перевірки пароля залежить від швидкості комп'ютера зловмисника і потужності словника паролів.

Є багато інструментів, які можуть бути використані зловмисником для експлуатації вразливостей мереж WiFi. Найбільш часто використовуваними є інструменти, зазначені нижче.

Aircrack-ng 802.11. Програма для підбору WEP і WPA-PSK ключів, яка може відновити ключі при достатній кількості перехоплених пакетів даних. Вона реалізує стандартну FMS (Fluhrer, Mantin and Shamir) атаку разом з деякими поліпшеннями, наприклад, KoreK-атакою, а також всі новими атаками PTW (Pyshkin, Tews, Weinmann), що робить процес підбору набагато швидше в порівнянні з іншими інструментами WEP зламу. Крім того, програма пропонує словниковий метод визначення ключа WEP. Для злому WPA / WPA2 Pre-Shared Key, використовується тільки словниковий метод [19].

Airmon-ng входить до складу пакету Aircrack-ng і використовується для включення і відключення режиму монітора для бездротових інтерфейсів. Вона також може бути використана для повернення з режиму монітора до режиму керування [19].

Airodump-ng включена в пакет Aircrack-ng і використовується для захоплення пакетів з вихідних 802.11 фреймів. Вона ідеально підходить для збору WEP IV(Initialization vector) при використанні з Aircrack-ng [19].

Aireplay-ng входить до складу пакету Aircrack-ng і використовується для інжекції бездротових фреймів. Її основна роль полягає в генерації трафіку для подальшого використання в Aircrack-ng для злому WEP і WPA-PSK ключів. Aireplay-ng

використовує різноманітні атаки, які можуть деаутентифікувати бездротові клієнти з метою отримання даних рукостискання WPA [19].

Besside-ng є інструментом, який підтримує WPA шифрування. Він в автоматичному режимі отримує несанкціонований доступ до WEP мереж у визначеному діапазоні і веде перелік рукостискань WPA [19].

Wireshark є загальновідомим аналізатором мережевих протоколів та має багатий набір функцій, який включає в себе наступне:

- можливість детального аналізу сотень протоколів,
- захоплення трафіку в режимі реального часу і автономний аналіз,
- захоплені мережеві дані можуть бути переглянуті за допомогою графічного інтерфейсу користувача або у TTY-режимі через TShark утиліти,
- підтримка дешифрування для багатьох протоколів, включаючи IPsec, ISAKMP, Kerberos, SNMPv3, SSL / TLS, WEP і WPA / WPA2 [19].

Всі ці інструменти можна знайти у складі широко доступних спеціальних зборок операційних систем на базі операційної системи Linux: Kali Linux, Parrot Security OS, Black Arch і т.д.

Таким чином, початкові умови експерименту включають в себе наступне:

- використання протоколу UDP,
- аутентифікація і шифрування передачі даних на основі специфікації WPA2-PSK,
- рівень кваліфікації зловмисника достатній для використання інструментів Aircrack-ng, Airmon-ng, Airodump-ng, Aireplay-ng, Besside-ng, Wireshark.

Передбачуваний результат експерименту полягає в заміні даних про температуру від датчика в інтерфейсі користувача на дані, що генеруються потенційним зловмисником з підробленого пристрою вимірювання на базі ESP32.

4. РЕЗУЛЬТАТИ ЕКСПЕРИМЕНТУ ТА ОБГОВОРЕННЯ

В експерименті використовувалася Parrot Security OS. Назва мережевої карти комп'ютера зловмисника, через який буде здійснюватися несанкціонований доступ WLAN1. Назва мережі Wi-Fi, через яку передаються дані про температуру від датчика до веб-інтерфейсу, STAARS. Для отримання пароля до мережі було використано словник паролів від RockYou, який є універсальним словником для отримання доступу до мережі Wi-Fi.

На першому етапі необхідно отримати несанкціонований доступ до STAARS. Щоб зробити це, треба виконати наступні дії:

- припинити всі процеси, які підключені до мережі (команда airmon-ng check kill),
- змінити режим WLAN1 на режим моніторингу (команда airmon-ng start wlan1),
- здійснити попередній перегляд всіх доступних точок доступу для вибору бажаної,
- створити інший термінал для продовження контролю мережевих пакетів,
- захопити рукостискання (Beside-ng), яке автоматично записується в файл wpa.cap.
- запустити процес підбору пароля.

ISSN 2663 - 4023

На другому етапі для перехоплення і аналізу мережевого трафіку було використано Wireshark (рис. 2).

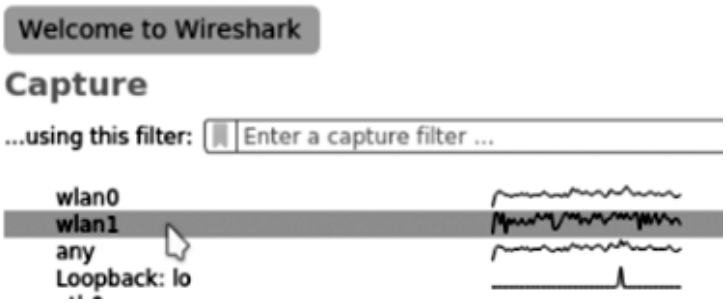

Рис. 2. Інтерфейс вибору мережі в Wireshark

На рис. 3 показана інформація, необхідна для відключення клієнта від сервера і підключення комп'ютера зловмисника до сервера, який передає дані про температуру і створення підробленого клієнту, який посилає не оригінальні дані.

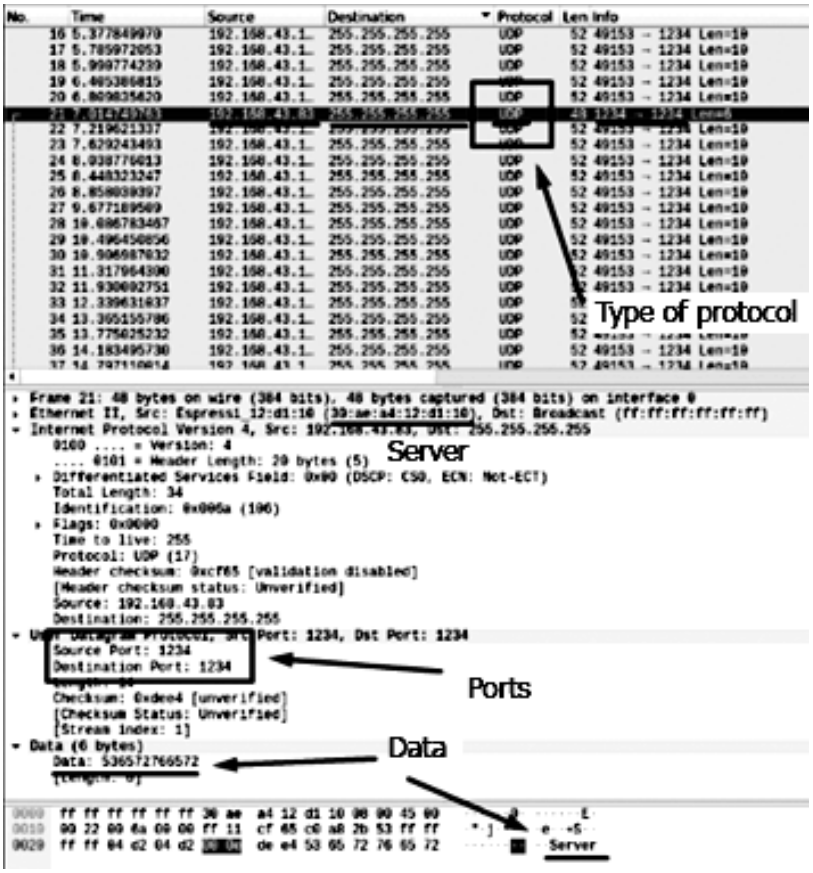

Рис. 3. Скріншот Wireshark інтерфейсу із зазначенням необхідних даних.

На третьому етапі було використано Arduino IDE для створення підробленого клієнта. На рис. 4 вказано код, який використовує інформацію, зібрану в Wireshark.

ISSN 2663 - 4023

Рис. 4. Приклад програмного коду для підробленого клієнта.

На четвертому етапі після запуску підробленого клієнта від сервера відключається оригінальний клієнт. Результати аналізу STAARS мережевого трафіку (рис. 5) показують, що підроблені дані передаються в мережу.

Рис. 5. Мережевий трафік STAARS

Таким чином, в результаті роботи може бути сформульований наступний сценарій атаки:

1. отримання несанкціонованого доступу до мережі (зміна режиму мережевої карти на режим моніторингу (Airmon-ng), перегляд доступних точок доступу (Airodump-ng), перехоплення рукостискання, вгадування пароля (Besside-ng);
2. перехоплення і аналізу мережевого трафіку (Wireshark);
3. створення підробленого клієнта ESP32 використовуючи отримані раніше дані (Arduino) і підключення його до мережі;
4. від'єднання оригінального пристрою на базі ESP32 від сервера (Aircrack-ng).

Як видно з описаного сценарію для першого, другого і четвертого етапів кваліфікація зловмисника повинна бути достатньою, щоб використовувати стандартні інструменти злому для роботи з бездротовими мережами. Дії на етапі 3 передбачають базові навички програмування ESP32. Однак, в цілому, описаний сценарій може бути реалізований зловмисником з низьким рівнем знань і навичок в галузі мережевих технологій і програмування ESP32. Це дозволяє досить легко такий сценарій реалізувати.

Слід зазначити, що у реалізованій конфігурації мережі Wi-Fi найуразливішою точкою є UDP протокол, адже він не вимагає перевірки стану переданих пакетів даних, що дозволяє зловмисникові їх перехопити і замінити. Відповідно, для того, щоб запобігти такому сценарію, достатньо використовувати протокол TCP, який на відміну від UDP, забезпечує цілісність даних і повідомлення відправника про результати передачі.

6. ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

Аналіз сучасних тенденцій в області IoT технологій дозволив визначити сегмент, для якого забезпечення інформаційної безпеки може зіткнутися з недостатністю її рівня. Це пристрої IoT на основі ESP32 мікроконтролера, які розроблені і впроваджені в домашніх умовах непрофесіоналами. У статті було обґрунтовано технічні рішення для створення фізичної моделі простої системи збору та передачі даних щодо температури навколишнього середовища до веб-інтерфейсу користувача. Реалізована модель була використана при проведенні експерименту з отримання несанкціонованого доступу до переданих даних в лабораторних умовах.

В результаті експерименту визначено конкретний сценарій атаки. Показано, що зловмисник, який має базові знання та навички в роботі із загальновідомими інструментами злому бездротових мереж (Aircrack-ng, Airmon-ng, Airodump-ng, Aireplay-ng, Besside-ng, Wireshark) і базові знання ESP32 і та навички програмування ESP32, може отримати доступ до системи і відправити підроблені дані в веб-інтерфейс. Для того, щоб зменшити ймовірність пропонованого сценарію рекомендується використовувати протокол TCP замість UDP.

Подальшим напрямком досліджень є тестування інших відомих проектів, які можуть бути реалізовані непрофесіоналами на базі ESP32. Зокрема включення до фізичної моделі актуаторів.

ISSN 2663 - 4023

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

[1] A. Colakovic and M. Hadžialic "Internet of Things (IoT): A review of enabling technologies, challenges, and open research issues", *Computer Networks*, vol. 144, pp. 17-39, 2018. DOI: https://doi.org/10.1016/j.comnet.2018.07.017

[2] Pathan A.K. *Securing Cyber-Physical Systems*. Boca Raton:CRC Press, 2015. DOI: https://doi.org/10.1201/b19311

[3] Misra S., Maheswaran M. and Hashmi S. *Security Challenges and Approaches in Internet of Things*. Springer, 2017. DOI: https://doi.org/10.1007/978-3-319-44230-3

[4] Aziz B., Arenas A. and Crispo B. *Engineering Secure Internet of Things Systems*. London:CPI Group, 2016. DOI: https://doi.org/10.1049/PBSE002E

[5] Gilchrist A. *Industry 4.0. The Industrial Internet Of Things*. Apress, 2016. DOI: https://doi.org/10.1007/978-1-4842-2047-4

[6] Hu F. *Security and Privacy in Internet of Things*. Boca Raton:CRC Press, 2016. DOI: https://doi.org/10.1016/B978-0-12-805395-9.00010-1

[7] Macaulay T. *RIoT Control. Understanding and Managing Risks and the Internet of Things*, Cambridge:Elsevier, 2017. DOI: https://doi.org/10.1016/B978-0-12-419971-2.00001-7

[8] Russell B. and Duren D. *Practical Internet of Things Security*. Birmingham:Packt Pub, 2016.

[9] Ibrahim D. *The Complete ESP32 Projects Guide. 59 Experiments with Adruino IDE and Python*. Elektor, 2019.

[10] Kurniawan A. *Internet of Things Projects with ESP32. Build exciting and powerful IoT projects using the all-new Espressif ESP32*. Birmingham:Packt Pub, 2019.

[11] A. Maier, A. Sharp, Y. Vagapov "Comparative Analysis and Practical Implementation of the ESP32 Microcontroller Module for the Internet of Things", *in 2017 Internet Technologies and Applications (ITA)*, Wales, UK, 2017, pp. 1-6. DOI: https://doi.org/10.1109/ITECHA.2017.8101926

[12] "20+ ESP32 Projects and Tutorials | Random Nerd Tutorials", *Random Nerd Tutorials*, 2019. [Online]. Available: https://randomnerdtutorials.com/projects-esp32/. [Accessed: 12- Aug- 2019].

[13] "The Internet of Things with ESP32", *Esp32.net*, 2019. [Online]. Available: http://esp32.net/. [Accessed: 12- Aug- 2019].

[14] *DS18B20 Programmable Resolution 1-Wire Digital Thermometer*. Maxim Integrated, 2018.

[15] V. Karagiannis, P. Chatzimisios, F. Vazquez-Gallego, J. Alonso-Zarate "A Survey on Application Layer Protocols for the Internet of Things", *Transaction on IoT and Cloud Computing*, vol. 1, no. 1, p. 1-10, 2015. DOI: https://doi.org/10.5281/zenodo.51613

[16] T. Salman, R. Jain "A Survey of Protocols and Standards for Internet of Things", *Advanced Computing and Communications*, Vol. 1, No. 1, p. 1-20, 2017.

[17] D. Costinela-Luminita "Wireless LAN Security - WPA2-PSK Case Study", *in 2nd World Conference on Information Technology (WCIT-2011)*, Antalya, Turkey, 2011, pp. 62-67.

[18] Salmon A., Levesque W. and McLafferty M. *Applied Network Security*. Birmingham:Packt Pub, 2017.

[19] *Tools.kali.org*, 2019. [Online]. Available: https://tools.kali.org/tools-listing. [Accessed: 12- Aug- 2019].

КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 2 (6), 2019
ISSN 2663 - 4023

Oleksii Barybin
Ph.D., Acting dean of Faculty of Physics and Technology
Vasyl' Stus Donetsk National University, Vinnytsia, Ukraine
ORCID ID 0000-0002-0897-4454
o.barybin@donnu.edu.ua

Elina Zaitseva
Ph.D., Associate professor of Computer Technology Department
Vasyl' Stus Donetsk National University, Vinnytsia, Ukraine
ORCID ID 0000-0002-2135-8146
zaytseva.elina@gmail.com

Volodymyr Brazhnyi
First year master`s program student of specialty 105 Applied Physics and Nanotechnology of Department of Radiophysics and Cyber-security
Vasyl' Stus Donetsk National University, Vinnytsia, Ukraine
ORCID ID 0000-0003-3327-9709
brazhnyi.v@donnu.edu.ua

TESTING THE SECURITY ESP32 INTERNET OF THINGS DEVICES

Abstract. In the paper analysis of current trends in IoT technologies enabled us to identify a segment for which information security assurance may encounter a lack of its level. These are IoT devices based on ESP32 microcontroller, that designed and implemented at home by non-professionals. The physical model of a handmade IoT system that includes device for measuring temperature based on ESP32 (small-sized ESP32-based development board ESP32 devKit V2 produced by Espressif and digital temperature sensor DS18B20), WiFi home network (based on router TL-WR841N) and web interface (based on node-red-dashboard) was proposed and implemented upon laboratory scale. The initial conditions of the experiment included the following: the use of the UDP protocol, authentication and data encryption based on WPA2-PSK specification, attacker skill level sufficient for use Aircrack-ng tools, Airmon-ng, Airodump-ng, Aireplay-ng, Besside-ng, Wireshark. The result of the experiment based on this model to attempt to gain unauthorized access to the transmitted data was successful. Attack scenario was formulated and consist of four stages: 1 – gaining unauthorized access to a network (network card transfers to monitor mode (Airmon-ng), view available access points (Airodump-ng), handshake interception, guessing the password (Besside-ng); 2 – network traffic interception and analysis (Wireshark); 3 – creating fake ESP32 client using the captured data (Arduino) and connect it to the network; 4 – disconnecting original ESP32 from a server (Aircrack-ng). It is shown that the attacker, who has the basic knowledge and skills in working with common wireless network hacking tools and a basic knowledge of ESP32 and ESP32 programming skills can access the system and send fake information to the web interface. To reduce the probability of the proposed scenario it is recommended to use TCP protocol, which is in contrast to UDP ensures data integrity and sender notification of transmission results.

Keywords: Internet of Things; attack scenario; ESP32; WiFi.

REFERENCES (TRANSLATED AND TRANSLITERATED)

[1] A. Colakovic and M. Hadžialic "Internet of Things (IoT): A review of enabling technologies, challenges, and open research issues", *Computer Networks*, vol. 144, pp. 17-39, 2018. DOI: https://doi.org/10.1016/j.comnet.2018.07.017

[2] Pathan A.K. *Securing Cyber-Physical Systems*. Boca Raton:CRC Press, 2015. DOI: https://doi.org/10.1201/b19311

[3] Misra S., Maheswaran M. and Hashmi S. *Security Challenges and Approaches in Internet of Things*. Springer, 2017. DOI: https://doi.org/10.1007/978-3-319-44230-3

[4] Aziz B., Arenas A. and Crispo B. *Engineering Secure Internet of Things Systems*. London:CPI Group, 2016. DOI: https://doi.org/10.1049/PBSE002E

[5] Gilchrist A. *Industry 4.0. The Industrial Internet Of Things*. Apress, 2016. DOI: https://doi.org/10.1007/978-1-4842-2047-4

[6] Hu F. *Security and Privacy in Internet of Things*. Boca Raton:CRC Press, 2016. DOI: https://doi.org/10.1016/B978-0-12-805395-9.00010-1

[7] Macaulay T. *RIoT Control. Understanding and Managing Risks and the Internet of Things*, Cambridge:Elsevier, 2017. DOI: https://doi.org/10.1016/B978-0-12-419971-2.00001-7

[8] Russell B. and Duren D. *Practical Internet of Things Security*. Birmingham:Packt Pub, 2016.

[9] Ibrahim D. *The Complete ESP32 Projects Guide. 59 Experiments with Adruino IDE and Python*. Elektor, 2019.

[10] Kurniawan A. *Internet of Things Projects with ESP32. Build exciting and powerful IoT projects using the all-new Espressif ESP32*. Birmingham:Packt Pub, 2019.

[11] A. Maier, A. Sharp, Y. Vagapov "Comparative Analysis and Practical Implementation of the ESP32 Microcontroller Module for the Internet of Things", *in 2017 Internet Technologies and Applications (ITA)*, Wales, UK, 2017, pp. 1-6. DOI: https://doi.org/10.1109/ITECHA.2017.8101926

[12] "20+ ESP32 Projects and Tutorials | Random Nerd Tutorials", *Random Nerd Tutorials*, 2019. [Online]. Available: https://randomnerdtutorials.com/projects-esp32/. [Accessed: 12- Aug- 2019].

[13] "The Internet of Things with ESP32", *Esp32.net*, 2019. [Online]. Available: http://esp32.net/. [Accessed: 12- Aug- 2019].

[14] *DS18B20 Programmable Resolution 1-Wire Digital Thermometer*. Maxim Integrated, 2018.

[15] V. Karagiannis, P. Chatzimisios, F. Vazquez-Gallego, J. Alonso-Zarate "A Survey on Application Layer Protocols for the Internet of Things", *Transaction on IoT and Cloud Computing*, vol. 1, no. 1, p. 1-10, 2015. DOI: https://doi.org/10.5281/zenodo.51613

[16] T. Salman, R. Jain "A Survey of Protocols and Standards for Internet of Things", *Advanced Computing and Communications*, Vol. 1, No. 1, p. 1-20, 2017.

[17] D. Costinela-Luminita "Wireless LAN Security - WPA2-PSK Case Study", *in 2nd World Conference on Information Technology (WCIT-2011)*, Antalya, Turkey, 2011, pp. 62-67.

[18] Salmon A., Levesque W. and McLafferty M. *Applied Network Security*. Birmingham:Packt Pub, 2017.

[19] *Tools.kali.org*, 2019. [Online]. Available: https://tools.kali.org/tools-listing. [Accessed: 12- Aug- 2019].