УДК 004.42, 004.49
DOI: 10.18523/2617-3808.2024.7.89-97

Глибовець А. М., Щербина С. С., Кирієнко О. В.

ВРАЗЛИВОСТІ БЕЗПЕКИ ТА РІШЕННЯ ДЛЯ ЗАХИСТУ В СИСТЕМАХ ІНТЕРНЕТУ РЕЧЕЙ

У статті представлено аналіз розробки комплексного рішення для захисту IoT систем та відомих і новітніх рішень у цій сфері.

Спочатку було окреслено шарове представлення архітектури IoT систем, а саме рівні сприйняття, мережі, оброблення даних і застосунку. Кожному з цих рівнів притаманні як спільні вразливості, так і унікальні. Ми уточнили критичні точки вразливості, охарактеризували основні проблеми автентифікації та авторизації. Зазначили, що стандартні облікові дані визначають як найпоширенішу та найпростішу складову вразливість, якою користуються зловмисники.

Проаналізовано наукові роботи, присвячені вирішенню проблем у сфері контролю доступу: централізовані центри довіри в TLS протоколі і пропозицію переходу на розподілені центри; випадки застосування IoT пристроїв без традиційних способів контролю доступу.

Значну увагу приділено шифруванню. Досліджені такі шифрувальні протоколи та методи, як TLS, DTLS, Novel Tiny Symmetric Encryption Algorithm, Lightweight CA Cipher (LCC) та Functional Encryption (FE), а також їхнє оптимальне застосування в IoT.

Ключові слова: Інтернет речей, вразливості безпеки, архітектура IoT, центр довіри, TLS протокол, DTLS, протокол, Novel Tiny Symmetric Encryption Algorithm, Lightweight CA Cipher (LCC) та Functional Encryption (FE).

Вступ

Інтернет речей розпочав трансформацію нашого цифрового простору, в якому повсякденні об'єкти є взаємопов'язаними та здатні до комунікації між собою. Ця трансформація не тільки спрощує нам життя, а й відкриває дорогу до небачених раніше можливостей та ефективності у сферах розумного будинку, медицини, індустріального виробництва, міського управління тощо [1]. Проте, як і з будь-яким технологічним проривом, IoT (Internet of Things) потребує зваженого та обережного впровадження. Насамперед, масштабне та неконтрольоване впровадження IoT пристроїв провокує багато проблем з безпекою, які необхідно вирішувати.

Основою IoT пристроїв є здатність збирати, оброблювати і передавати інформацію автономно, без людського втручання. Хоча саме це і є головною інновацією, така автономність створює багато вразливостей у безпеці. Наприклад, IoT пристрої часто використовують як приватні, так і публічні мережі, таким чином розширюючи потенційний вектор атаки зловмисника на конфіденційність і цілісність даних. Але це не єдине, на що потрібно зважати. Не менш важливими є надійність і доступність критично важливого функціоналу, який такі прилади надають. Від термостатів і годинників до автономних транспортних засобів та інфраструктури міста, вплив скомпрометованих IoT пристроїв може варіюватися від простої незручності до катастрофічних збоїв, що вплинуть на життя мільйонів людей.

IoT прилади зазвичай мають мало обчислювальної потужності та оперативної пам'яті. Такі обмеження призводять до того, що часто такі традиційні підходи до безпеки, як складні алгоритми шифрування, неможливо впровадити. Зважаючи на це і на вибухове зростання кількості IoT приладів у використанні, також збільшується і потенціал для несанкціонованого втручання в роботу систем. За прогнозом Cybersecurity Ventures, кількість IoT пристроїв у світі зросте до 25,1 мільярда до 2025 року. Кожний із них є потенційною точкою входу для зловмисника. Таке значне поширення створює велику, часто погано захищену мережу взаємопов'язаних приладів, яку можуть використати зловмисники для власних цілей: крадіжки даних, DDoS атак тощо.

Більше того, non-patchable (що не підлягають виправленню) та forever-day (вічні) вразливості тільки ускладнюють проблему. Багато IoT пристроїв нечасто отримують оновлення прошивки через логістичні проблеми доставлення цих оновлень або ж тому, що виробник про них забув і більше не підтримує. Такі вразливості являють собою постійно відкриті двері для зловмисників, і тому потрібен інноваційний підхід до поточних рішень.

Інфраструктуру навколо IoT пристроїв можливо розділити на чотири рівні [2]: сприйняття (захоплення вузла, імперсоніфікація вузла, атака відтворення, часова атака, атака позбавлення сну), мережі (підслуховування, DoS/DDoS атака, Main-in-The-Middle, оброблення даних (виснаження ресурсів, експлойти), застосунку (експлойти). Отже, основними проблемами для безпеки в IoT є несанкціонований доступ до пристроїв і даних, підслуховування комунікації, маніпуляція комунікації, виснаження ресурсів систем і пристроїв.

Метою цієї статті є опис запропонованого комплексного рішення для захисту IoT систем. Існує нагальна потреба в надійних і здатних до масштабування рішень. Наш підхід враховує унікальні характеристики IoT екосистем, а саме різноманітність, обмеженість в обчислювальних ресурсах і фізичну природу приладів.

Проблеми автентифікації та авторизації

IoT прилади мають багато вразливостей безпеки. Серед них можливо виокремити використання слабких паролів, паролів за замовчуванням і hardcoded (вписаних у прошивку пристроїв) паролів. Ця вразливість вирізняється поміж інших через свою поширеність і простоту. Автентифікація та авторизація є критично важливими для забезпечення безпеки IoT приладів. Автентифікація відповідає за те, щоби тільки дозволені прилади та користувачі мали змогу отримати доступ до сервісу. Авторизація ж — контролює дії, які може робити авторизована сутність. У контексті IoT, традиційних підходів може бути недостатньо,

Важливими тут є поняття довіри та центрів довіри. У роботі [7] запропоновано використовувати локальну розподілену систему управління довірою. Завдяки децентралізації процесів автентифікації та авторизації і розміщенню їх ближче до місця, де генеруються та обробляються дані, IoT може досягти вищого рівня безпеки. Цей метод вирішує проблеми масштабу та різноманітності в системах IoT, зменшує затримку, підвищує ефективність і потенційно пропонує більшу стійкість проти атак.

Зрозуміло, що потрібно окрему увагу надавати IoT пристроям, у яких немає традиційних інтерфейсів користувача, таких як клавіатура і сенсорний екран [13]. Як приклад вразливостей, пов'язаних із нетрадиційними способами автентифікації та авторизації таких пристроїв, наведено Apple Watch, із його функціоналом розблокування за допомогою розблокування зв'язаного з ним телефону. Якщо такий годинник було втрачено, власник, нічого не підозрюючи, може розблокувати свій телефон, а з ним і годинник. Далі зловмисник може отримати доступ до Mac жертви за допомогою годинника. Тож автори стверджують про потребу так званої постійної автентифікації та авторизації.

Динамічний характер середовищ IoT, де пристрої можуть непостійно перебувати під контролем користувача, потребує переходу від одноразових механізмів автентифікації до рішень, які можуть постійно перевіряти особу користувача. Також розглядають варіанти з використанням унікальних біологічних властивостей людини, таких як ритм серцебиття, кількість світла чи радіохвиль, що відбиває людське тіло тощо.

За допомогою машинного навчання, гіпотетично, це можливо перетворити на системи автентифікації та авторизації.

Стаття [5] пропонує вирішення проблеми захисту IoT пристроїв у середовищах, де традиційні заходи безпеки можуть бути неможливими, наприклад у зонах з обмеженими каналами під'єднання і ресурсами. Найперше, це запис унікальних облікових даних у прошивку на стадії виробництва. Такий метод є серйозною перепоною для імперсоніфікації пристрою чи підслуховування комунікації, проте потребує більше зусиль на стадії виробництва та підтримки. Дієвим є і варіант, коли облікові дані пристроїв зберігаються у вигляді фізичних QR кодів і додаються до сервера авторизації перед тим, як їх віддадуть користувачам. QR код залишається в безпечному місці. В цьому і подальшому варіантах використовують так звану довіру через фізичну близькість, а QR код застосовують для процесу ACE обміну. Так, облікові дані пристроїв можуть зберігатися у вигляді фізичних QR кодів і додаватися до сервера авторизації напряму перед використанням. У цьому варіанті існує проблема втрати QR коду та подальшої компрометації. Також, якщо QR було скомпрометовано, його

неможливо змінити без зміни прошивки. Життєвим є варіант, коли випадково згенерований QR код, що надано з пристроєм, використовують для початкового шифрування та отримання нового ключа. Цей процес можливо виконати тільки після фізичного перезавантаження пристрою, щоби уникнути використання «довіри через фізичну близькість» як вразливість.

Для забезпечення безпечної комунікації використовують CoAP через IPv6 із DTLS. Також важливою частиною всієї системи є можливість відкликати токени авторизації (token revocation). Пристрої та ресурс сервер постійно запитують у сервера авторизації інформацію про актуальність токенів та доступів.

Шифрування

У величезному та взаємопов'язаному світі IoT шифрування даних є гарантом конфіденційності та цілісності. Оскільки пристрої IoT збирають, передають і обробляють щораз більший обсяг конфіденційних даних, брак надійного шифрування даних стає явною вразливістю. Це не тільки наражає користувачів на ризик витоку даних і конфіденційності, а й підриває довіру, необхідну для ширшого впровадження IoT. Зважаючи на це й унікальну природу IoT приладів, а саме — часту обмеженість у ресурсах, виникає потреба в обчислювально легких і достатньо надійних методах та парадигмах шифрування.

Протокол TLS можна вважати стандартним вибором для впровадження захисту комунікації. Наприклад, TLS використовують в інфраструктурах, що містять брокер повідомлень MQTT. TLS (transport layer security) у своїй основі має протокол SSL (Secure Sockets Layer), розроблений компанією Netscape [6]. Протокол було реалізовано над транспортним рівнем, у цьому випадку – над TCP. Свого часу правильно сконфігурований SSL надавав хороші гарантії безпеки. Потенційний зловмисник міг дізнатися лише параметри з'єднання, такі як конкретний тип шифрування, частота обміну та кількість даних.

Протокол TLS виконує такі функції: шифрування, автентифікація і гарантія цілісності даних. Задля встановлення безпечного зв'язку учасники мають вибрати методи шифрування та ключі. В протоколі цей процес має назву TLS Handshake. Протокол використовує парадигму криптографії з відкритим ключем. Це означає, що клієнти можуть встановити безпечний зв'язок без попередніх знань один про одного, проте це залежить від обраного криптографічного набору (наприклад, для TLS-SRP потрібно попередньо знати про спільний секрет). Також у процедурі TLS Handshake підтверджується автентичність презентованих сторін, зазвичай клієнт перевіряє сервер, проте існує можливість взаємної перевірки, відома як mTLS.

Кожне надіслане повідомлення містить MAC (Message Authentication Code) код, що створюється за допомогою односторонньої криптографічної функції хешування та виконує функцію контрольної суми. Цю контрольну суму також генерує і отримувач для визначення цілісності інформації.

Ще у версії SSL 2.0 було перебачено скорочений процес TLS Handshake, що має назву abbreviated handshake. В пакеті ServerHello сервер може надіслати 32-байтний ідентифікатор сесії. Цей ідентифікатор генерується на сервері та зберігається в кеш для подальшого використання. Клієнт, отримавши ідентифікатор сесії, також зберігає його в себе. Тепер, якщо з'єднання було розірвано, клієнт може в пакеті ClientHello надіслати вищезгаданий ідентифікатор. Якщо учасники з'єднання мають однакові ідентифікатори, процес TLS Handshake буде скорочено. Якщо ні — виконається повна версія.

Abbreviated handshake дає змогу пропустити генерацію симетричного ключа, що суттєво скорочує час встановлення з'єднання. Проте за такого підходу сервер має зберігати дані про всі минулі сесії, що на великих обсягах може впливати на його швидкодію. Для вирішення цієї проблеми існує механізм Session Ticket. Якщо клієнт підтримує такий механізм, то під час першого з'єднання замість ідентифікатора сесії сервер надішле Session Ticket — параметри сесії, зашифровані приватним ключем сервера. В такому варіанті при відновленні з'єднання клієнт замість ідентифікатора сесії надсилає вищезгаданий Ticket, що дозволяє серверу не зберігати дані про сесію.

Механізм Abbreviated Handshake пришвидшує процес відновлення сесії, проте він ніяк не впливає ні на найперший етап встановлення зв'язку, TLS Hadnshake, ні на випадок, коли сесія вже не є дійсною. Щоби пришвидшити роботу і тут, існує опціональне розширення TLS False Start. Воно дає змогу надсилати дані ще до завершення процесу TLS Hadnshake. В результаті використання такого механізму стає на одну ітерацію обміну повідомленнями менше.

Із різних причин раніше частіше за все обмін ключами відбувався за допомогою алгоритму RSA [9]. Цей алгоритм має один недолік: якщо зловмисник заволодіє приватним ключем сервера, він зможе розшифровувати всі сеанси зв'язку з цим сервером. Навіть гірше, зловмисник може роками записувати шифрований трафік, маючи на меті розшифрувати дані, коли він отримає приватний ключ сервера. На відміну від RSA, процес обміну ключами Діффі-Геллмана (D-H) [11] є більш захищеним, бо вибраний симетричний ключ шифрування ніколи не виходить за межі ні клієнта, ні сервера, і тому не може бути перехоплений по мережі. DH зменшує ризики, пов'язані з компрометацією попередніх сеансів зв'язку, оскільки для кожного нового сеансу створюється новий тимчасовий симетричний ключ.

Варто ще раз зауважити, що шифрування з публічним ключем використовується тільки в процесі TLS Handshake. Після встановлення зв'язку дані шифруються і розшифровуються за допомогою симетричної криптографії. Такий підхід вибрано для збільшення швидкодії, оскільки шифрування з публічним ключем є набагато більш вимогливим у розрізі обчислювальних ресурсів.

У протоколі TLS важливим концептом є ланцюг довіри. Коли клієнт звертається до сервера, він має мати змогу підтвердити автентичність наданого публічного ключа / сертифіката. Зробити це він може декількома способами, а саме: certiticate/public key pinning — клієнт має збережений сертифікат або публічний ключ сервера, який він порівнює з наданим; перевірка наданого сертифіката за допомогою сертифіката центру довіри (CA certificate).

Ланцюг довіри основано на сертифікатах автентичності, що їх надають спеціальні установи — центри сертифікації (CA, certificate authorities). З виданих сертифікатів складається певний ланцюг. Причиною використання ланцюга є розподілення ризиків безпеки. Сертифікати містять інструкції з перевірки їхньої актуальності, тому під час перевірки сертифіката потрібно перевіряти весь ланцюг довіри. В основі процедури перевірки лежить Certificate Revocation List (CRL) — список відкликаних сертифікатів. Кожний центр сертифікації має такий список. Якщо сертифікат там згадано, то це означає, що його було відкликано і з'єднання встановлювати не можна. Проте запит на отримання всього списку відкликаних сертифікатів для перевірки одного є нераціональним із погляду ресурсів. До того ж, такі списки публікують періодично, а не постійно. Тому було розроблено механізм Online Certificate Status Protocol (OCSP), що дає змогу перевіряти статус сертифіката динамічно й селективно.

Найбільшою проблемою використання TLS для деяких приладів буде обмеженість обчислювальних ресурсів. Проте існує велика кількість полегшених бібліотек та імплементацій [3], які залишають лише найнеобхідніші компоненти протоколу, позбавляючись всього іншого. Наприклад, можливо створити TLS бібліотеку, що займатиме лише 20 Кб оперативної пам'яті під час роботи. Більшість складнощів при розробці таких рішень полягає у великій кількості підтримуваних шифрувальних комплектів. Якщо взяти за основу TLS 1.2 комплект TLS_RSA_WITH_AES_128_CBC_SHA256, то доведеться імплементувати тільки RSA, AES та SHA-256. Також процес Handshake можливо зробити синхронним і позбавитися всіх додаткових механізмів, як False Start та Abbreviated Handshake. Найскладнішою частиною всієї імплементації є робота з X.509 сертифікатами, але і тут можливо спростити, використавши механізм certificate pinning. Інший варіант — використати бандл TLS_SRP_SHA_WITH_AES_128_CBC_SHA, але тоді треба мати наперед відомий спільний секрет. Варто ще раз зауважити, що процес обміну RSA не надає forward secrecy захисту, тому, все ж таки, рекомендують використовувати ECDHE шифр.

Datagram Transport Layer Security (DTLS) — протокол, побудований над UDP, що загалом надає ті самі гарантії безпеки і має схожу структуру, що і TLS. Використовують у таких протоколах, як CoAP. Через використання UDP як транспортного протоколу DTLS має як плюси, так і мінуси порівняно з TLS. Найчастіше застосовують у системах із великим обсягом стримінгових даних. DTLS дає змогу уникати повільної комунікації, що характерна для TLS. Проте, водночас, він не змінює порядок пакетів і не гарантує їхньої унікальності (non-replayability). Всі відмінності між TLS та DTLS пов'язані з використанням протоколу UDP [4].

Крім шифрування, DTLS також використовують для запобігання втраті або надходженню пакетів даних у неправильному порядку. DTLS використовує для цього простий таймер повторного передання, при цьому кожний учасник зв'язку продовжує повторно передавати своє останнє повідомлення, доки не буде отримано відповідь. TLS ділить довгу послідовність даних на кілька фрагментів. Протокол є відповідальним за розбір цих фрагментів на кінці отримувача, і тому клієнтський код не потребує жодних додаткових модифікацій. Натомість DTLS використовує записи, які можна надсилати повністю чи ні. Цими записами повинні керувати самі програми, тобто протокол не є відповідальним

за це. Так само, як і UDP, у DTLS немає сигналу завершення передання. У DTLS передання даних просто припиняється. Це означає, що клієнт, який отримує дані від сервера, не знає, чи було доставлено всі дані, чи сталася помилка під час зв'язку. TLS сигналізує про це сповіщенням.

DTLS використовує cookie для запобігання підміні IP-адреси. Це дає змогу спілкуватися із сервером, не розкриваючи клієнтську IP-адресу. З іншого боку, TLS здійснює зв'язок лише після встановлення TCP-handshake, що ускладнює підміну IP-адреси.

У дослідженні SuHyun Kim і ImYeong Lee [8] запропоновано використовувати Proxy Re-Encryption (PRE) — перешифрування на проксі. Цей спосіб шифрування вирішує критичну проблему захисту передання даних в екосистемах IoT, що характеризується безліччю взаємопов'язаних пристроїв з обмеженими обчислювальними можливостями.

PRE — це криптографічна парадигма, яка дає змогу делегувати права на дешифрування. Вона надає проксі ноді можливість перетворювати дані, шифровані ключем A, на дані, які можливо розшифрувати ключем B. Під час цієї операції дані у відкритому вигляді не з'являються.

Цей механізм особливо підходить для середовища IoT з кількох причин. Пристрої IoT часто мають обмежену обчислювальну потужність і пам'ять, що в цих випадках робить традиційні схеми шифрування непрактичними. PRE пропонує полегшену альтернативу, яка враховує ці обмеження. Інфраструктура IoT часто вимагає безпечного обміну даними між кількома пристроями або користувачами. PRE полегшує це, дозволяючи безпечно та безпосередньо ділитися зашифрованими даними без необхідності розшифровки та повторного шифрування, заощаджуючи таким чином обчислювальні ресурси. IoT-ландшафт дуже динамічний, де пристрої часто приєднуються до мережі та виходять з неї. PRE підтримує цей динамізм, забезпечуючи гнучке та безпечне керування ключами та механізми контролю доступу до даних.

У контексті IoT PRE можна використовувати так. У шифруванні кожний пристрій генерує собі пару ключів: публічний (pk) і приватний (sk). Потім він генерує ключ перешифрування, щоби обмінюватися даними з іншими пристроями. Щоби пристрій A міг поділитися даними з пристроєм B, йому потрібно згенерувати ключ повторного шифрування rk(A→B). Він генерується за допомогою sk(A) та pk(B) і далі надсилається на проксі ноду. У цьому сценарії кожен пристрій створює ключ перешифрування для всіх пристроїв, крім себе. Проксі ноду перешифровує дані від пристрою A за допомогою ключа rk(A→B) та надсилає на пристрій B. Пристрій B розшифровує дані за допомогою sk(B).

У статті [10] запропоновано новий підхід до TEA шифрування. Цей метод спрямований на підвищення безпеки та ефективності передання текстових файлів між пристроями IoT, що відповідає критичній потребі в легких криптографічних рішеннях.

NTSA розроблено для подолання обмежень алгоритму Tiny Encryption Algorithm (TEA) і його варіантів. Вони, не зважаючи на невелике використання пам'яті та простоту імплементації, страждають від вразливостей безпеки через використання постійного ключа протягом усього процесу шифрування. NTSA вводить динамічну плутанину ключів (Dynamic Key Confusions) для кожного раунду шифрування, значно покращуючи безпеку зашифрованих даних. Цей підхід особливо корисний для середовищ Інтернету речей, де пристрої часто працюють із суворими обмеженнями ресурсів.

На відміну від TEA, який використовує той самий ключ для всіх раундів шифрування, NTSA динамічно генерує додаткові ключі для кожного раунду. Динамічні ключі генеруються на основі основного ключа та даних, що планується зашифрувати. Цей метод значно підвищує безпеку процесу шифрування, збільшуючи складність ключа та роблячи його більш стійким до атак. Окрім того, NTSA спеціально оптимізовано для безпечної передачі текстових файлів у системах IoT. Хоча NTSA і пропонує розширені гарантії безпеки, цей метод шифрування залишається легким і ефективним. Це робить його придатним для використання у пристроях IoT, де кількість обчислювальних можливостей є обмеженою.

У статті [12] представлено шифрування за назвою Lightweight Cellular Automata Based Encryption Technique — полегшене шифрування на основі клітинних автоматів (CA).

LCC використовує принципи клітинного автомата — математичної моделі, яка складається з сітки комірок, кожна з яких перебуває в одному зі скінченної кількості станів, наприклад «увімкнено» або «вимкнено». Стан комірки на наступному кроці часу визначається набором правил на основі поточного стану комірки і станів сусідніх комірок. LCC використовує простоту та ефективність скінченного автомата, щоб забезпечити надійний, але легкий метод шифрування, який підходить для обмежених середовищ пристроїв IoT. Цей метод шифрування є симетричним.

Алгоритм LCC розроблено так, щоб бути ефективним як із погляду обчислювальних вимог, так і з точки зору споживання енергії. Це, своєю чергою, робить його особливо корисним для використання в IoT пристроях, бо вони часто живляться від батарей і мають обмежені ресурси для оброблення даних.

У методі LCC ключем шифрування слугує наперед обраний набір правил RVList512 і кількість ітерацій (1–8). Вводом слугує 512 біт початкових даних.

Правила клітинних автоматів можуть бути згенеровані залежно від радіуса та кількості можливих значень, які клітина може мати як значення стану. У випадку одновимірного клітинного автомату з радіусом r = 1 (три можливі сусіди (лівий, центральний, правий)), можна створити 256 можливих правил CA, але усі ці комбінації не утворюють правила GCA (груповий клітинний автомат). Для генерації GCA правил використовують алгоритм, який випадковим чином бере вісім правил CA(RV8) із доступних 256 правил CA, а потім генерує GCA вектор (RVList512) шляхом конкатенації. Такий вектор, що містить 512 правил CA, можна використовувати в шифруванні. Зловмисник має витратити майже експоненційний час, щоби вгадати 512 правил, що зводить нанівець можливість брутфорс атаки.

Функціональне шифрування (Functional Encryption, FE) є узагальненням наявних технологій шифрування за допомогою відкритого ключа, такі як шифрування на основі ідентифікації (IBE), на основі атрибутів (ABE), гомоморфне шифрування (HE), шифрування предикатів (PE), шифрування з можливістю пошуку (SE) тощо [14].

FE являє собою зміну парадигми технології шифрування, що дає змогу більш детально контролювати доступ до зашифрованих даних. У системі FE ключі дешифрування видаються на основі певних функцій; власники цих ключів можуть обчислювати лише певні функції над зашифрованими даними, не вивчаючи нічого іншого про дані. У застосуванні в IoT FE пропонує новий підхід до захисту конфіденційних даних клієнтів, водночас дозволяючи постачальникам послуг виконувати необхідну аналітику та оброблення даних.

Функціональне шифрування дає змогу точно контролювати, які дані можна розшифрувати й обробити, залежно від дозволів, пов'язаних із ключем розшифрування. Воно гарантує, що конкретні актори можуть отримати доступ лише до тих даних, на які вони мають право. Кожен фахівець може отримати доступ лише до даних, що стосуються його галузі, забезпечуючи конфіденційність клієнта.

Стратегії реагування на інциденти безпеки

Безпека IoT інфраструктури є критичним аспектом, що потребує комплексного підходу. З огляду на різноманітність атак і вразливостей потрібно впроваджувати багаторівневий підхід до безпеки, що передбачає як технічні, так і організаційні рішення. Основними проблемами для безпеки в IoT є несанкціонований доступ до пристроїв і даних, підслуховування комунікації, маніпуляція комунікації, виснаження ресурсів систем і пристроїв.

Також необхідно враховувати загрози, що є специфічними для кожного окремо рівня інфраструктури, та впроваджувати відповідні заходи безпеки, наприклад: використання надійних методів шифрування, регулярне оновлення програмного забезпечення, моніторинг мережевої активності, впровадження горизонтальної архітектури для систем, фізичний захист IoT пристроїв тощо.

Проблему, яку становлять вразливі облікові дані в IoT приладах, неможливо переоцінити. Усунення цієї вразливості потребує зусиль від виробників, користувачів і навіть державних установ.

Забезпечення захисту IoT пристроїв і мереж, у яких вони оперують, потребує імплементації надійних механізмів автентифікації та авторизації. Враховуючи часто конфіденційний характер даних і потенційні наслідки дір у безпеці, розроблення і впровадження таких механізмів мають першочергове значення. Рішення мають бути ефективними та масштабованими, здатними адаптуватися до унікальних вимог IoT екосистем.

У світі IoT шифрування даних відіграє ключову роль у захисті конфіденційності та цілісності інформації. Через зростання обсягу чутливих даних, які збирають і обробляють пристрої IoT, надійне шифрування є критично важливим для запобігання витокам даних і порушенням конфіденційності. Це особливо актуально у зв'язку з обмеженістю ресурсів IoT пристроїв, що підвищує необхідність у легких і водночас ефективних методах шифрування. У статті ми розглянули як перевірені часом, так і нові підходи, які можуть бути використані для підвищення безпеки IoT систем. Викла-

дені методи та парадигми забезпечують міцну основу для створення надійних, ефективних і безпечних систем, здатних адаптуватися до різних вимог.

Враховуючи складність і різноманітність середовищ Інтернету речей, традиційні стратегії реагування на інциденти часто виявляються неефективними. Для ефективної та правильної реакції на випадки компрометації систем Інтернету речей потрібен спеціальний підхід.

Неоднорідність і велика різноманітність пристроїв Інтернету речей створюють значні проблеми для моніторингу, виявлення та реагування. Це означає, що різні групи пристроїв потенційно можуть потребувати різного підходу для вищезгаданих процесів. Отримання повного логу операцій пристроїв і відкритого мережевого трафіку є складним завданням, що ускладнює виявлення та розуміння масштабу інцидентів. Обмежена потужність обчислювальних ресурсів і кількість постійної пам'яті багатьох пристроїв Інтернету речей обмежують обсяг даних, які можна зібрати після інциденту. Крім того, потенційна фізична недоступність деяких пристроїв ускладнює аналіз, якщо є підозра на фізичний характер інциденту порушення безпеки.

Висновок

Метою цієї роботи був аналіз розробки комплексного рішення для захисту IoT систем і відомих і новітніх рішень у цій сфері.

Спочатку було окреслено шарове представлення архітектури IoT систем, а саме рівні сприйняття, мережі, оброблення даних та застосунку. Кожному з цих рівнів притаманні як спільні вразливості, так і унікальні. Ми уточнили критичні точки, які можуть використати зловмисники для отримання доступу до систем чи перешкоджання роботи тих чи тих вузлів.

Потім ми сфокусували увагу на проблемах автентифікації та авторизації. Зауважили, що стандартні облікові дані визначаються як найпоширеніша та найпростіша складова вразливість, якою користуються зловмисники. Ця вразливість існує в великому переліку систем і вже не один раз призводила до широких успішних кібератак, як-от Mirai ботнет.

Далі аналізували наукові роботи, присвячені менш поширеним проблемам у сфері контролю доступу: проблемі централізованих центрів довіри в TLS протоколі і пропозиції переходу на розподілені центри; випадкам застосування IoT пристроїв без традиційних способів контролю доступу чи з потребою в продовжуваному контролі доступу; використанню IoT у несприятливих умовах (зоні бойових дій), де існує ризик фізичної втрати пристроїв чи серверів і компрометації систем, та способам захисту в таких сценаріях.

Значну увагу приділено шифруванню. Тут досліджено такі шифрувальні протоколи та методи, як TLS, DTLS, Novel Tiny Symmetric Encryption Algorithm, Lightweight CA Cipher (LCC) та Functional Encryption (FE), а також їхнє оптимальне застосування в IoT.

Унікальні характеристики IoT екосистем потребують спеціального підходу до управління загрозами та реагування на інциденти. Розуміючи конкретні загрози, з якими стикаються пристрої та мережі IoT, і запроваджуючи систему реагування на інциденти, адаптовану до цих проблем, організації можуть підвищити свою стійкість проти кібератак. Це передбачає не лише технічні заходи, а й організаційну готовність, зокрема навчання, планування та аналіз після інциденту. Завдяки старанним зусиллям і стратегічному плануванню можна значно посилити безпеку всієї інфраструктури, захищаючи як пристрої, так і дані, які оброблюються пристроями.

Список літератури

1. Глибовець А. М. Агентно-базовані програмні системи пошуку та аналізу інформації / А. М. Глибовець. — Київ : Видавничий дім «Києво-Могилянська академія», 2019. — 281 с. : іл.
2. Burhan M. IoT Elements, Layered Architectures and Security Issues: A Comprehensive Survey / M. Burhan, R. Rehman, B. Khan, B.-S. Kim // Sensors. —2018. — Vol. 18, no. 9. — P. 2796. — https://doi.org/10.3390/s18092796.
3. Computationally simple, lightweight replacement for SSL/TLS Information Security Stack Exchange, Apr. 2011 [Electronic resource]. — Mode of access: https://security.stackexchange.com/questions/3204/computationally-simple-lightweight-replacement-for-ssl-tls (date of access: 18.09.2024).
4. Difference TLS Vs DTLS protocol [Electronic resource] // The Network DNA. — 2022. — Mode of access: https://www.thenetworkdna.com/2022/11/difference-tls-vs-dtls-protocol.html (date of access: 18.9.2024).
5. Echeverría S. Authentication and Authorization for IoT Devices in Disadvantaged Environments / S. Echeverría, G. A. Lewis, D. Klinedinst, L. Seitz // 2019 IEEE 5th World Forum on Internet of Things (WF-IoT). — Limerick, Ireland, 2019. — Pp. 368–373. — https://doi.org/10.1109/WF-IoT.2019.8767192.
6. Grigorik I. 4. Transport Layer Security (TLS) — High Performance Browser Networking [Electronic resource] / I. Grigorik. — Mode of access: https://www.oreilly.com/library/view/high-performance-browser/9781449344757/ch04.html (date of access: May 26, 2024).

7. Kim H. Authentication and Authorization for the Internet of Things / H. Kim, E. A. Lee // IT Professional. — 2017. — Vol. 19, no. 5. — Pp. 27–33. — https://doi.org/10.1109/mitp.2017.3680960.
8. Kim S. IoT device security based on proxy re-encryption / S. Kim, I. Lee // Journal of Ambient Intelligence and Humanized Computing. — 2017. — Vol. 9, no. 4. — Pp. 1267–1273. — https://doi.org/10.1007/s12652-017-0602-5.
9. Nemec M. The Return of Coppersmith's Attack: Practical Factorization of Widely Used RSA Moduli / M. Nemec, M. Sys, P. Svenda, D. Klinec, V. Matyas // 24th ACM Conference on Computer and Communications Security (CCS'2017). ACM: 1631–1648. — https://doi.org/10.1145/3133956.3133969.
10. Rajesh S. A Secure and Efficient Lightweight Symmetric Encryption Scheme for Transfer of Text Files between Embedded IoT Devices / S. Rajesh, V. Paul, V. Menon, M. Khosravi // Symmetry. — 2019. — Vol. 11, no. 2. — P. 293. — https://doi.org/10.3390/sym11020293.
11. Rescorla E. Diffie-Hellman Key Agreement Method [Electronic resource] / E. Rescorla // RFC 2631 — 1999. — Mode of access: https://datatracker.ietf.org/doc/html/rfc2631.
12. Roy S. A Lightweight Cellular Automata Based Encryption Technique for IoT Applications / S. Roy, U. Rawat, J. Karjee // IEEE Access. — 2019. — Vol. 7. — Pp. 39782–39793. — https://doi.org/10.1109/ACCESS.2019.2906326.
13. Shahzad M. Continuous Authentication and Authorization for the Internet of Things / M. Shahzad, M. P. Singh // IEEE Internet Computing. — 2017. — Vol. 21, no. 2. — Pp. 86–90. — https://doi.org/10.1109/MIC.2017.33.
14. Sharma D. Functional Encryption in IoT E-Health Care System / D. Sharma, Devesh Jinwala // Lecture notes in computer science. — 2015. — Pp. 345–363. — https://doi.org/10.1007/978-3-319-26961-0_21.

References

Burhan, M., Rehman, R., Khan, B., & Kim, B.-S. (2018). IoT Elements, Layered Architectures and Security Issues: A Comprehensive Survey. *Sensors*, *18* (9), 2796. https://doi.org/10.3390/s18092796.

Computationally simple, lightweight replacement for SSL/TLS. (April, 2011). Information Security Stack Exchange. https://security.stackexchange.com/questions/3204/computationally-simple-lightweight-replacement-for-ssl-tls.

Difference TLS Vs DTLS protocol. (2022, 4 November). *The Network DNA*. https://www.thenetworkdna.com/2022/11/difference-tls-vs-dtls-protocol.html.

Echeverría S., Lewis G. A., Klinedinst, D., and Seitz, L. (2019). Authentication and Authorization for IoT Devices Disadvantaged Environments. In *IEEE 5th World Forum on Internet of Things (WF-IoT), Limerick, Ireland* (pp. 368–373). https://doi.org/10.1109/WF-IoT.2019.8767192.

Grigorik, I. (2013). 4. Transport Layer Security (TLS*).* In *High Performance Browser Networking. O'Reilly Online Learning.* https://www.oreilly.com/library/view/high-performance-browser/9781449344757/ch04.html.

Hlybovets, A. (2019). *Ahentno-bazovani prohramni systemy poshuku ta analizu informatsii*. Vydavnychyi dim "Kyievo-Mohylianska akademiia".

Kim, H., & Lee. E. A. (2017a). Authentication and Authorization for the Internet of Things. *IT Professional, 19* (5), 27–33. https://doi.org/10.1109/mitp.2017.3680960.

Kim, S., & Lee, I. (2017b). IoT device security based on proxy re-encryption. *Journal of Ambient Intelligence and Humanized Computing*, *9* (4), 1267–1273. https://doi.org/10.1007/s12652-017-0602-5.

Nemec, M., Sys, M., Svenda, P., Klinec, D., & Matyas, V. (2017). The Return of Coppersmith's Attack: Practical Factorization of Widely Used RSA Moduli. In *24th ACM SIGSAC Conference on Computer and Communications Security*. ACM. https://doi.org/10.1145/3133956.3133969.

Rajesh, S., Paul, V., Menon, V. G., Khosravi, M. R. (February, 2019). A Secure and Efficient Lightweight Symmetric Encryption Scheme for Transfer of Text Files between Embedded IoT Devices. *Symmetry*, *11* (2), 293. https://doi.org/10.3390/sym11020293.

Rescorla, E. (June, 1999). Diffie-Hellman Key Agreement Method. *RFC 2631 https://datatracker.ietf.org/doc/html/rfc2631*.

Roy, S., Rawat, U., & Karjee, J. (2019). A Lightweight Cellular Automata Based Encryption Technique for IoT Applications. *IEEE Access*, *7*, 39782–39793. https://doi.org/10.1109/ACCESS.2019.2906326.

Shahzad, M., & Singh, M. P. (2017). Continuous Authentication and Authorization for the Internet of Things. *IEEE Internet Computing*, *21* (2), 86–90. https://doi.org/10.1109/MIC.2017.33.

Sharma, D., & Jinwala Devesh. (January, 2015). Functional Encryption in IoT E-Health Care System. *Lecture notes in computer science*, 345–363. https://doi.org/10.1007/978-3-319-26961-0_21.

A. Hlybovets, S. Shcherbyna, O. Kyriienko

SECURITY VULNERABILITIES AND PROTECTION SOLUTIONS IN INTERNET OF THINGS SYSTEMS

The Internet of Things (IoT) has begun transforming our digital space, in which everyday objects are interconnected and capable of communicating with each other. This transformation not only simplifies our lives but also creates unprecedented opportunities and enhances efficiency in areas such as smart homes, healthcare, industrial manufacturing, and urban management. However, as with any technological breakthrough, IoT requires a careful and well-planned implementation. The large-scale and unregulated deployment of IoT devices raises significant security concerns that must be mitigated.

At the core of IoT devices lies the ability to autonomously collect, process, and transmit information independently of human intervention. While this autonomy is the primary innovation, it also introduces numerous security vulnerabilities. For instance, IoT devices often operate on private and public networks, increasinf the attack surface for malicious actors capable of compromising data confidentiality and

integrity. However, security is not the sole concern. The reliability and availability of the critical functions these devices provide are equally crucial. From thermostats and smartwatches to autonomous vehicles and urban infrastructure, compromised IoT devices can cause anything from minor inconveniences to catastrophic failures affecting millions.

IoT devices typically have limited computational power and memory. These constraints often render the implementation of traditional security measures, such as complex encryption algorithms, impractical. Given these limitations and the explosive growth in the number of IoT devices, the potential for unauthorized interference with systems also increases. According to Cybersecurity Ventures, the number of IoT devices worldwide is expected to reach 25.1 billion by 2025. Each device represents a potential entry point for attackers. This widespread proliferation creates a vast, often poorly secured, network of interconnected devices susceptible to exploitation by malicious actors for data theft, DDoS attacks, and other cyber threats.

The infrastructure surrounding IoT devices can be divided into four levels: perception (node capture, node impersonation, replay attack, timing attack, sleep deprivation attack), network (eavesdropping, DoS/DDoS attacks, Man-in-the-Middle attacks), data processing (resource exhaustion, exploits), and application (exploits). Therefore, key security challenges in IoT include unauthorized access to devices and data, communication interception, data manipulation, and depletion of system and device resources.

The purpose of this article is to describe a comprehensive proposed solution for securing IoT systems. There is an urgent need for scalable and reliable solutions. Our approach considers the unique characteristics of IoT ecosystems, particularly their diversity, limited computational resources, and the physical nature of the devices.

Keywords: Internet of Things, security vulnerabilities, IoT architecture, trust center, TLS protocol, DTLS, protocol, Novel Tiny Symmetric Encryption Algorithm, Lightweight CA Cipher (LCC), Functional Encryption (FE).

Матеріал надійшов 27.09.2024