ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЗАХИСТУ ІНФОРМАЦІЇ

КАФЕДРА ІНФОРМАЦІЙНОЇ ТА КІБЕРНЕТИЧНОЇ БЕЗПЕКИ

ПОЯСНЮВАЛЬНА ЗАПИСКА

до магістерської роботи
на тему:

«ТЕХНОЛОГІЇ ЗАХИСТУ ПРИСТРОЇВ ІОТ НА БАЗІ ПРОТОКОЛУ LORA WAN»

Виконала: студентка 6 курсу, групи БСДМ-61
Спеціальності 125 Кібербезпека
Освітньо-професійної програми «Інформаційна та кібернетична безпека»
(шифр і назва спеціальності)
Маслова Ю.Ю.
(прізвище та ініціали)
Керівник Гайдур Г.І.
(прізвище та ініціали)
Рецензент _____
(прізвище та ініціали)
Нормоконтролер Чумак Н.С.
(прізвище та ініціали)

КИЇВ – 2020

РЕФЕРАТ

Текстова частина магістерської роботи: 72 сторінки, 5 рисунків, 29 джерел.

Об'єкт дослідження: технології захисту пристроїв IOT на базі протоколу LoRaWAN.

Предмет дослідження: об'єкт інформаційної діяльності, пристроїв IOT на базі протоколу LoRaWAN.

Мета роботи: дослідження технологій захисту пристроїв IOT на базі протоколу LoRaWAN під час різних етапів життєвого циклу.

Методи дослідження: сукупність теоретичних та практичних методів для проведення аналізу загроз та технології захисту пристроїв IOT на базі протоколу LoRaWAN.

В роботі проведено аналіз життєвого циклу пристроїв IOT, цикл розробки ПЗ IOT, проаналізовано загрози та технології захисту пристроїв, а також протокол LoRaWAN, способи взаємодії пристроїв, архітектуру мережі та способи захисту інформації в рамках зазначеного протоколу.

Перша частина висвітлює саме стандарт LoRaWAN способи взаємодії пристроїв, архітектуру мережі та способи захисту інформації в рамках зазначеного протоколу.

В другій частині досліджені основні етапи життєвого циклу пристроїв IOT, етапи розробки ПЗ IOT в контексті захисту інформації та проаналізовані загрози що можуть виникати на різних стадіях життєвого циклу пристроїв IOT.

В третій частині досліджені проблеми та висунуті рекомендації та пропозиції щодо організації процесів взаємодії з зацікавленими особами, безпосередньо, щодо процесів отримання певного пристрою IOT в контексті безпеки та оглянуті технології захисту пристроїв IOT на базі протоколу LoRaWAN.

Практична частина роботи полягає у виявленні загроз та виборі технологій захисту пристроїв IOT на базі протоколу LoRaWAN

Напрямки подальших досліджень полягають у перевірці ефективності розробленої методики на практиці.

Галузь використання – інформаційна безпека організації.

ТЕХНОЛОГІЇ, LoRaWAN, LoRa, ПРОЦЕСИ, ЗАГРОЗИ, IOT, ЖИТТЄВИЙ ЦИКЛ.

ЗМІСТ

ВСТУП

Актуальність дослідження. В наш час, одним з ключових напрямків розвитку мереж зв'язку є концепція Інтернету речей [1, 2, 3]. За переважною більшістю прогнозів, "Інтернет речей" продовжить бурхливо розвиватися, та відповідно до прогнозів, в 2023 році кількість підприємств та корпорацій у відсотковому відношенні, що будуть використовувати IOT рішення сягне 70%. Кількість самих кінцевих пристроїв в цьому сегменті досягне позначки в кілька мільярдів.

Світ стоїть на порозі четвертої промислової революції. Так, відповідно до огляду, підготовленому до форуму в Давосі, Інтернет речей входить в топ-5 технологічних драйверів четвертої промислової революції [4, p. 7]. За даними всесвітнього дослідження PwC Digital IQ, IoT займає перше місце серед технологій, здатних змінити бізнес-моделі компаній або цілих індустрій [5].

Широкий спектр галузей, де є потреби в використанні IoT, а також різноманітність пристроїв і датчиків, що підключаються до мережі, збільшують попит в фахівцях, які знаються на таких системах і здатних з ними працювати. Збільшення популярності IoT призведе до нового буму на ринку праці. Тому напрямок захисту рішення на основі IoT фактично є надиктованим нашим часом.

У цій роботі розглянуто специфікацію LoRa WAN, класи кінцевих пристроїв, варіант побудови мережі, проаналізовані найвірогідніші загрози,

Об’єкт дослідження: технології захисту пристроїв IOT на базі протоколу LoRaWAN.

Предмет дослідження: об'єкт інформаційної діяльності, пристроїв IOT на базі протоколу LoRaWAN.

Мета роботи: дослідження технологій захисту пристроїв IOT на базі протоколу LoRaWAN під час різних етапів життєвого циклу.

Наукові завдання:

дослідити сутність проблеми захисту пристроїв IOT на базі протоколу LoRaWAN;

проаналізувати технології захисту на різних етапах життєвого циклу пристроїв IOT на базі протоколу LoRaWAN;

Методи дослідження: сукупність теоретичних та практичних методів для проведення аналізу загроз та технологій захисту пристроїв IOT на базі протоколу LoRaWAN.

1. ПРОТОКОЛ LoRaWAN

В березні місяці 2015 року Semtech Corporation зробили заяву про нове, і важливе досягнення в сфері технологій бездротової передачі даних. Вони презентували мережевий енергоефективний протокол LoRaWAN (Long Range Wide Area Networks).

Цей протокол забезпечує масу переваг на відміну від Wi-Fi і стільниковими мережами, завдяки застосуванню технологій M2M (міжмашинної комунікацій). На ринку бездротового зв'язку технологія викликала величезний інтерес у виробників і вендорів. Для її підтримки, розвитку та стандартизації створений альянс LoRa (LoRa Alliance). В даний час альянс розвивається, кількість його членів постійно збільшується. Станом на 2020 рік альянс складався більш ніж з 500 компаній. Членами альянсу стали компанії з виробництва пристроїв, технологій і різних сервісів. В склад альянсу увійшли всесвітньо відомі виробники електроніки: Cisco, IBM, Kerlink, IMST, Semtech, Microchip Technology, - а також провідні телекомунікаційні оператори (Bouygues Telecom, KPN, SingTel, Proximus, Swisscom).

Була сформульована основна задача альянсу: Об'єднана апаратне і програмне забезпечення на базі стандарту LoRaWAN. Це дозволить операторам зв'язку надавати послуги Інтернету речей комерційним організаціям і приватним фірмам. Використання зазначеного стандарту дасть можливість в значній мірі спростити з'єднання зростаючої кількості пристроїв. Технологія LoRa відноситься до класу LPWAN дає можливість проникати сигналу вглиб приміщень в місті, а також стабільно забезпечить покриття зони в сільській місцевості, або на підприємстві. Все це дасть можливість реалізувати, розробляти і вдосконалювати на базі LoRa різні додатки для проектів - “Розумне місто” або “Розумне підприємство”.

Semtech Corporation - один із засновників LoRa Alliance, а також виробник своїх пристроїв LoRa і бездротової радіочастотної технології LoRa і мереж LoRaWAN.

LoRaWAN - це відкритий протокол для мереж, які володіють високою ємністю (близько 1 мільйона пристроїв), низьким енергоспоживанням і великим радіусом дії (до 15 км на відкритій місцевості). Даний протокол забезпечує зв'язок між вузлами мережі і використовує особливі методи шифрування, що забезпечує надійність і безпеку системи.

Втілення сценарію розвитку Інтернету речей є концепція розвитку Розумного міста. Це амбітна містобудівна концепція, що припускає оптимізацію і раціоналізацію управління всіма міськими системами. Основним об'єктом “Розумного міста” є розумне ЖКГ. Всі елементи цього комплексу пов'язані: це і теплопостачання, газопостачання, електропостачання, бази техобслуговування і т.ін. Контроль за всіма системами міста стає можливим завдяки застосуванню нових технологій. Саме технологія LoRa допоможе вирішити безліч завдань з обслуговування “Розумного міста”. Що до концепції “Розумне підприємство” це може бути і розумний склад, і розумні виробничі лінії, іншими словами — це напрямок отримання трігерів стану з одних пристроїв та відповідне керування технологічним процесом за допомогою інших.

1.1 Принцип роботи мережі LoRaWAN

Стандарт LoRaWAN на ринку мережевих додатків з’явився нещодавно, але вже є маса прикладів його застосування. На рис. 1.1 показано, як станція LoRa - Інтернет речей проводить збір даних з кінцевих вузлів. Ці вузли за допомогою шлюзів утворюють невидимі мости, вони з'єднуються з центральним сервером. кінцеві вузли належать абонентам, а центральний сервер і шлюзи контролює оператор.

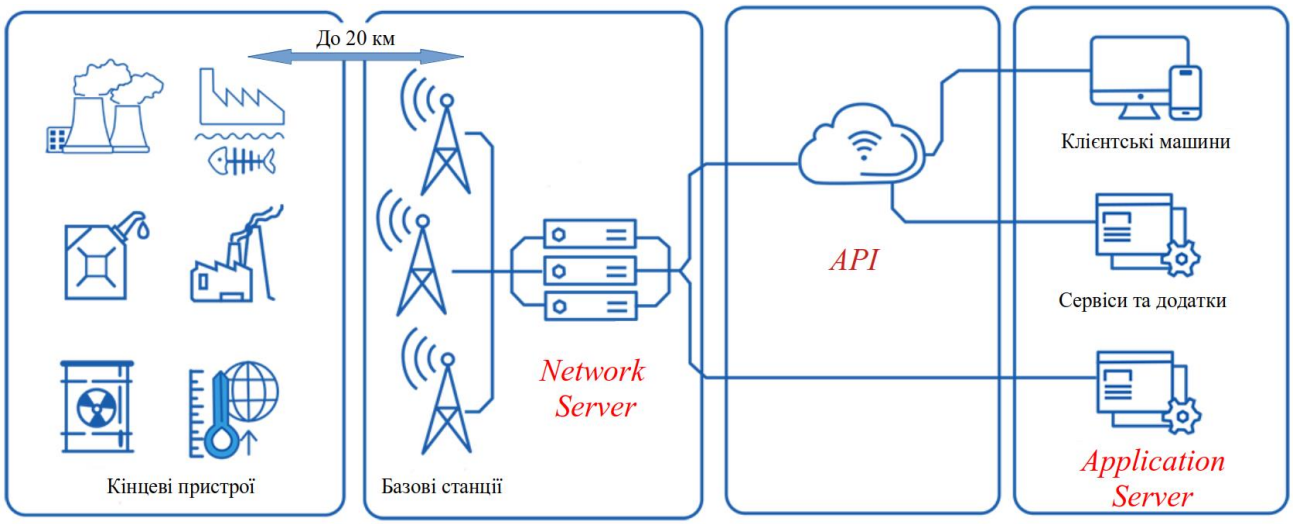

Рис. 1.1. Типова бездротова мережа LoRaWAN

LoRaWAN це глобальна мережа. Тому головне завдання розробників забезпечити захист даних користувачів. Для цього проводиться кодування на декількох рівнях: на мережевому рівні, наскрізна безпека на рівні додатків.

Типова бездротова мережа LoRaWAN являє собою сукупність шлюзів (gateways), які пересилають повідомлення між кінцевими пристроями (end-devices) і центральним сервером (Network Server, NS), і характеризується «зоряної» топологією «star-of-stars». Відображення отриманих метрик відбувається на сервері візуалізації (Application Server, AS), з’єднання NS та AS відбувається за допомогою IP-з'єднання.

У мережі LoRaWAN передбачено використання трьох класів пристроїв. Вони використовуються для вирішення різних завдань в залежності від області застосування.

- Двонаправлені кінцеві пристрої «класу А» (Bi-directional end- devices, Class A). Ці кінцеві пристрої застосовуються, коли потрібна мінімальна споживана потужність при передачі даних на сервер.

- Двонаправлені кінцеві пристрої «класу Б» (Bi-directional end- devices, Class B). Відмітна особливість від класу А - додаткове вікно прийому. Його

пристрій відкривається за розкладом. Це означає що передача даних з сервера буде здійснюватися тільки тоді, коли кінцеве пристрій вийде на зв'язок. Складання розкладу для кінцевого пристрою здійснить синхронізацію по сигналу від шлюзу.

- Двонаправлені кінцеві пристрої «класу С» (Bi-directional end- devices, Class C). У цих пристроїв максимальне вікно прийому. вони призначені для отримання великого обсягу даних і мають майже безперервне вікно прийому даних. Таким чином, 1 шлюз мережі обслуговує, та може керувати близько 5 тисяч кінцевих пристроїв.

Базова станція — типове поняття для багатьох радіосистем, включаючи мережі IoT. У мережі LoRaWAN базова станція (БС) виконує функції сполучення і взаємодії радіомережі з абонентським терміналом і концентрації навантаження з групи терміналів, тому в документації LoRa Alliance БС іменується шлюзом або концентратором. Сигнал від одного кінцевого пристрою або терміналу може прийматися декількома БС. Сукупність базових станцій оператора забезпечує радіопокриття мережі і прозору двосторонню передачу даних між кінцевими пристроями та сервером мережі. Базова станція оснащена приймально — передавальної антеною (секторної або всюди направленною), а також (опціонально) GPS / ГЛОНАСС — антеною для прецизійної синхронізації внутрішнього годинника і визначення точних координат приймально-передавальної антени.

Зв'язок шлюзів і центральним сервером відбувається як і між NS та AS, через стандартні IP-з'єднання а між шлюзами і кінцевими пристроями через бездротові з'єднання. Весь процес призначений для низько швидкісної бездротової передачі даних в неліцензійних діапазонах частот на великі відстані. Зв'язок є двосторонньої, але основний обсяг даних передається від кінцевих пристроїв до шлюзів.

Мережевий сервер (*NS*)- програмно-апаратний комплекс, який виконує наступні функції:

- Управління радіомережею. Мережевий сервер мережі LoRaWAN вибирає БС для передачі повідомлень в напрямку «вниз» (downlink), приймає рішення про необхідність зміни швидкості передачі даних для кожного терміналу, потужності передавача, контролює заряд батарей кінцевих пристроїв, шифрує дані і т.ін.

- Контроль радіомережі включає моніторинг, збір статистики і аварійне інформування.

- Маршрутизація пакетів даних від кінцевих пристроїв або терміналів до відповідних серверів додатків. Кожен пакет даних, що відправляється абонентським терміналом, має в своєму складі унікальний ідентифікатор DevAddr, а на мережевому сервері зберігається запис про відповідність DevAddr і URL сервера додатків, якій призначена інформація від терміналу кінцевого пристрою. На підставі цієї відповідності мережевий сервер виконує маршрутизацію пакета до сервера додатків, де відбувається його подальша обробка додатком.

Сервер додатків, сервер візуалізації - платформа, яка виконує перший рівень шифрування / дешифрування і виробляє обробку даних, одержуваних від кінцевих пристроїв, терміналів і направляються до терміналів або кінцевих пристроїв. Крім роботи з даними, сервер додатків може управляти кінцевими пристроями з рівня додатки (наприклад, переводити їх в режим роботи іншого класу, управляти опцією адаптивної передачі даних, мультикаст і т.ін.).

1.2 Забезпечення захисту даних в мережі LoRaWAN рівнем інфраструктури

Насамперед потрібно звернути увагу на те, яким чином можуть бути захищені кінцеві пристрої на етапі виготовлення а також на етапі під'єднання до мережі та подальшого обміну повідомленнями.

При виготовленні в абонентський пристрій заносяться наступні константи:

- DevEUI (Device Identifier - унікальний ідентифікатор пристрою [64 біта]);
- AppEUI (Application Identifier - унікальний ідентифікатор додатки [64 біта]). Починаючи з версії стандарту V1.1 називається JoinEUI - ідентифікатор Join-сервера;
- AppKey (Application Key - унікальний кореневої ключ шифрування [128 біт]). Використовується в процесі обчислення сесійних ключів шифрування AppSKey і NwkSKey.
- Активація пристрою (рис. 1.2) може здійснюватися як через радіоефір OTAA (рис. 1.3), так і оператором ABP (рис. 1.4). Під час процесу активації формуються наступні дані:
- DevAddr (End-Device Address - локальна адреса пристрою в конкретній мережі [32 біта]). Складається з двох полів: NetID і NwkAddr. Чим коротше NetID (ідентифікатор мережі оператора, який присвоюється LoRa Alliance, мінімальний розмір - 6 біт), тим довше поле NwkAddr і тим більше різних пристроїв може бути зареєстровано в мережі (максимум на один NetID - 33,5 млн терміналів, кінцевих пристроїв);
- NwkSKey (Network Session Key - мережевий сесійний ключ шифрування [128 біт]). Використовується для шифрування даних між терміналом і сервером, а також для обчислення поля MIC (Message Integrity Code) з метою перевірки цілісності даних при передачі їх по радіоефіру;

• AppSKey (Application Session Key - сесійний ключ шифрування додатка [128 біт]). Служить для шифрування даних на рівні додатку (між абонентським терміналом і сервером додатка).

Сесійні ключі шифрування формуються незалежно на абонентському терміналі і на мережевий частини на підставі ідентифікаторів мережі (NetID) і пристрої (DevEUI), кореневого ключа безпеки AppKey, а також згенерованих унікальних для кожної сесії випадкових чисел DevNonce, JoinNonce. При активації терміналу оператором зміна сесійних ключів шифрування без демонтажу пристрою неможлива, через що цей варіант активації використовувати не рекомендується.

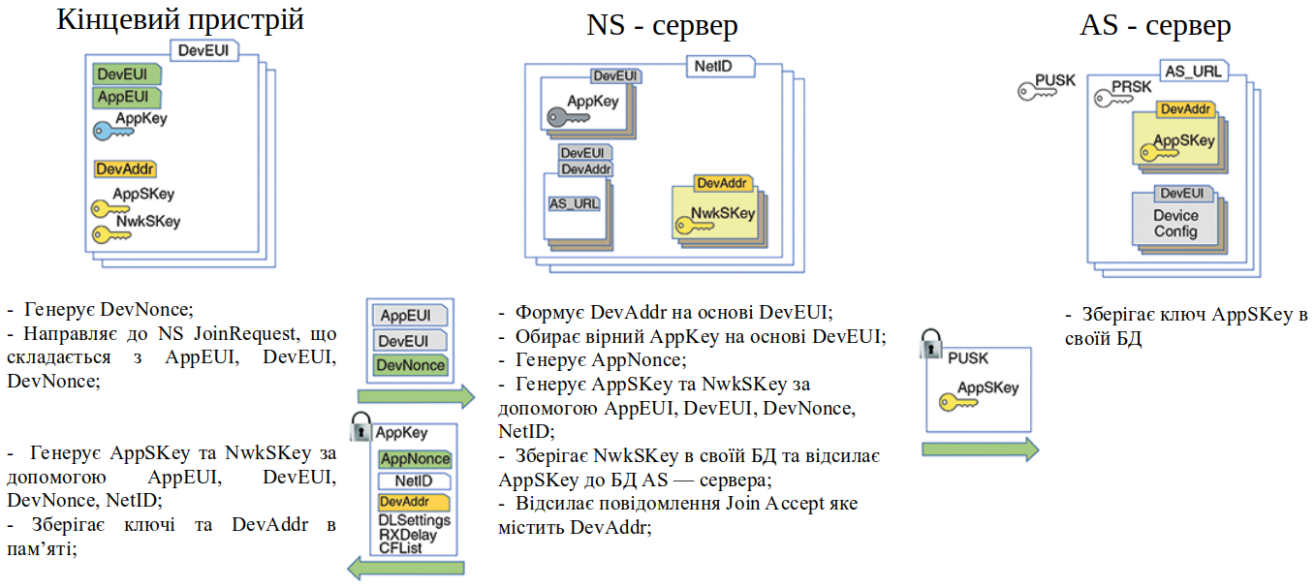

Рис. 1.2. Процедура активації кінцевого пристрою за допомогою радіоефіру (OTAA)

DevAddr, NwkSKey і AppSKey зберігаються в пам'яті абонентського пристрою, DevAddr, NwkSKey - на мережевому сервері (NS), а AppSKey - на сервері візуалізації, додатків (AS).

В загальному випадку активація за допомогою OTAA приведена на рис. 1.3.

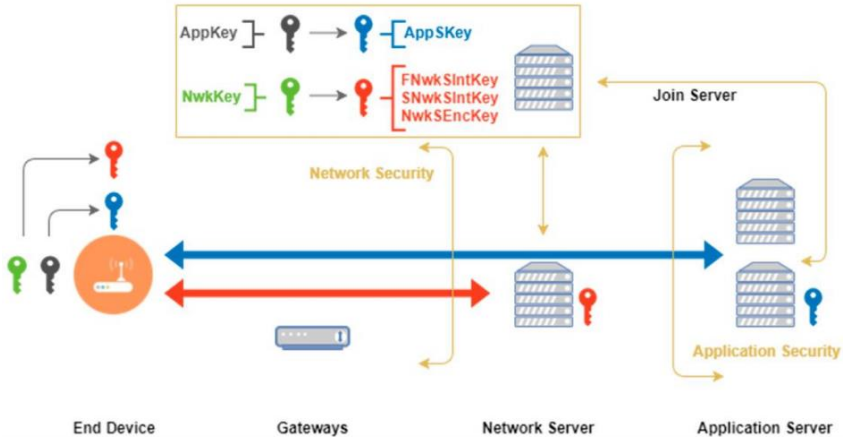

Рис. 1.3. Загальна схема активації пристрою за допомогою OTAA

У випадку активації за допомогою персоналізації (ABP) пристрою не потрібно DevEUI, AppEUI або AppKey. Замість цього ключі сеансу NwkSKey і AppSKey попередньо встановлені у вашому пристрої, і пристрій попередньо зареєстровано в мережі. Коли пристрій хоче встановити зв'язок, воно робить це, використовуючи ключі сеансу, без необхідності спочатку використовувати процедуру з'єднання. Загальна схема активації пристрою за допомогою ABP приведена на рис. 1.4.

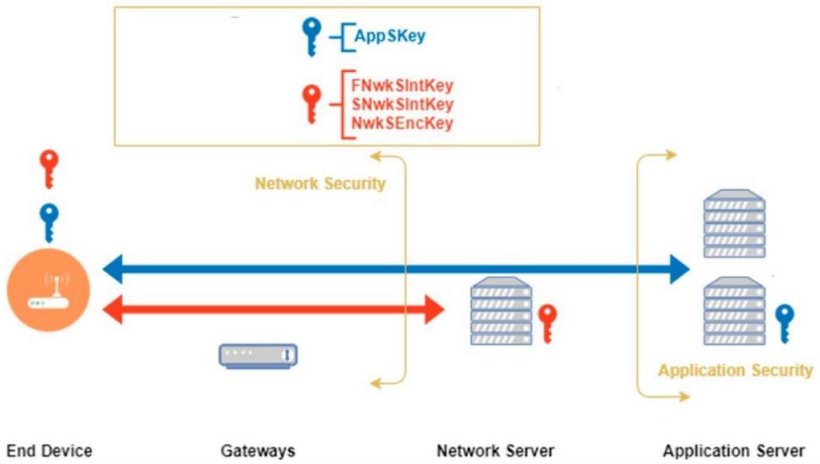

Рис. 1.4. Загальна схема активації пристрою за допомогою ABP

1.3 Шифрування повідомлень в мережі що працює за протоколом LoRaWAN

У мережі IoT LoRaWAN використовується багаторівнева система безпеки передачі даних (Рис. 1.5).

Перший рівень має AES — шифрування на рівні додатку (між кінцевим пристроєм і сервером додатків, візуалізації) за допомогою 128 — бітного змінного сесійного ключа Application Session Key (AppSKey). Цей ключ шифрування зберігається в абонентському терміналі і на сервері додатків і недоступний оператору зв'язку (доступ до AppSKey є тільки у клієнта — власника сервера додатків, візуалізації). Формування сесійного ключа AppSKey відбувається паралельно в абонентському терміналі і на стороні мережі під час процедури активації кінцевого пристрою, через повітря (під час ефіру) AppSKey не передається.

Другий рівень передбачає AES — шифрування і перевірку цілісності повідомлень на мережевому рівні (між абонентським терміналом і сервером) за допомогою 128 — бітного змінного сесійного ключа Network Session Key (NwkSKey). Зазначений ключ шифрування також зберігається в абонентському терміналі і на мережевому сервері і недоступний клієнту (доступ до NwkSKey є тільки у оператора мережі - власника мережевого сервера). Формування сесійного ключа NwkSKey також відбувається паралельно в абонентському терміналі і на стороні мережі під час процедури активації терміналу, через ефір NwkSKey також не передається.

Третій рівень включає стандартні методи аутентифікації і шифрування інтернет-протоколу IPsec, TLS і т.ін. При передачі даних по транспортній мережі між вузлами мережі (базова станція, мережевий сервер, join-сервер, сервер додатків).

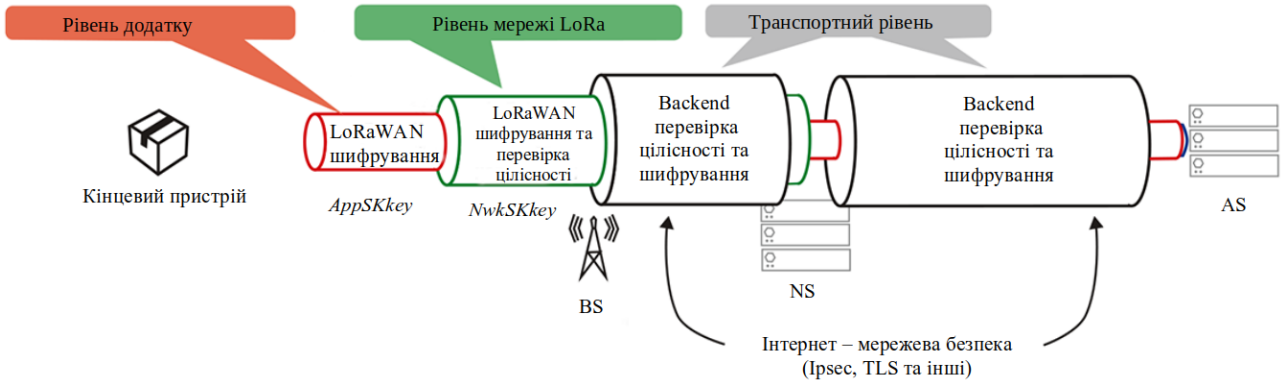

Рис. 1.5. Загальна схема безпеки даних в мережі LoRaWAN

За командою додатки або мережевого сервера в будь-який момент можливий перехід на нову сесію з генерацією нового комплекту ключів шифрування, що робить марними старі ключі шифрування. Також є можливість встановити періодичну генерацію нового комплекту ключів NwkSKey і AppSKey.

У версії стандарту LoRaWAN V1.0.x формування сесійних ключів на стороні мережі проводиться на мережевому сервері (NS), однак у версії V1.1 для цих цілей використовується виділений сервер - так званий join-сервер. Join-сервер може бути додатково захищений окремим апаратним модулем безпеки HSM (Hardware Security Module).

В цьому випадку для безпечної передачі згенерованих сесійних ключів між серверами, а також для зберігання їх на мережевому сервері і сервері додатків впроваджуються додаткові ключі: AS_Key для ключа AppSKey і LRC_K для ключа NwkSKey. На абонентському пристрої ключі шифрування за потреби можуть захищатися спеціальним апаратним елементом безпеки Secure Element, що виключить їх компрометацію в разі фізичного впливу на термінал.

Впровадження апаратних засобів захисту в мережі і на терміналі робить марними спроби перехоплення сесійних ключів при передачі їх між серверами і спроби зламу мережевих серверів або абонентських пристроїв з метою отримання сесійних ключів.

Розглянуті заходи також створюють умови для захищеного роумінгу даних - безпечну авторизацію датчиків в гостьовій мережі і захищену передачу даних домашньому сервера додатків з гостьової мережі.

З метою додаткового захисту процесу генерації сесійних ключів Join — сервер може бути розміщений на території клієнта або виробника пристроїв. У цьому випадку навіть співробітники оператора не зможуть отримати доступ до сесійних і кореневих ключів шифрування.

1.4 Забезпечення безпеки даних в мережі що працює за протоколом LoRaWAN від най розповсюджених загроз.

Перехоплення призначених для користувача даних

Найпростіша загроза - звичайне перехоплення даних.

Принцип реалізації атаки:

Через те, що радіохвилі поширюються неконтрольовано, будь хто може взяти приймач, налаштований на потрібний діапазон і тип модуляції, і слухати все, що передається від кінцевого пристрою до базової станції, шлюзу.

Спосіб захисту - шифрування даних.

Механізм реалізації:

У LoRaWAN призначені для користувача дані шифруються по алгоритму AES-128 з ключем довжиною, відповідно, 128 біт (16 байт).

Повторне відправлення даних

У LoRaWAN до кожного пакету додається лічильник. Якщо на сервер мережі приходить пакет з лічильником, рівним або менше попереднього, то цей пакет просто відкидається. Після переповнення від пристрою прийде пакет з номером 0, який, очевидно, буде менше будь-якого іншого номера, але при цьому сервер

мережі повинен його сприйняти правильно і почати відлік пакетів з нуля. Крім того, пристрій може обнулити лічильник перезавантаженням.

Це досягається в одним з наступних способів:

- перед відсиланням пакету пристрій повинен пройти процедуру реєстрації в мережі (Join)
- сервер допускає прихід чергового пакета з номером 0, при цьому відлік починається заново

У LoRaWAN використовуються обидві схеми в залежності від способу активації пристрою - OTAA або ABP.

Для OTAA використовується перший варіант, при цьому пристрою видаються нові ключі шифрування.

Для ABP, використовується другий варіант - втім, якщо термін переповнення лічильника помітно перевищує розрахунковий термін служби пристрою, він може бути відключений. На випадок випадкового перезавантаження, після відправлення кожного пакета таке кінцеве пристрій зберігає значення лічильника в незалежну пам'ять.

Друга схема, звичайно, менш безпечна, але на практиці допустима - зловмиснику в ній треба записати не будь-який пакет, а конкретно нульовий.

Підробка лічильника

Якщо лічильник підробити, можна покласти його в зашифровану частину пакета, але тоді реальний обсяг призначених для користувача даних зменшиться на два байта. Можна шифрувати не тільки призначені для користувача дані, але тоді, по-перше, доведеться підлаштуватися під 16 - байтний розмір блоку, а по - друге, збільшиться навантаження на сервер мережі, якому для будь-яких дій над пакетом доведеться його спочатку розшифровувати (в схемі ж, коли шифруються тільки призначені для користувача дані, якщо пакет за формальними ознаками ігнорується, то розшифровувати не треба нічого).

При цьому очевидно, що нам неважливо, знає зловмисник номер пакета чи ні - в схемі з перереєстрацією в мережі (OTAA) це знання йому взагалі нічим не допоможе, а в ABP він буде дуже довго чекати наступного приходу пакету з номером N-1. Тому цілком достатньо не дати йому цей номер змінити.

Для цього весь пакет в LoRaWAN підписується криптографічного підписом AES — CMAC, цей підпис в стандарті називається MIC, Message Integrity Code. По ньому перевіряється, що весь пакет, включаючи всі заголовки і дані, дійшов до сервера в незмінному вигляді.

Тобто, прийнявши черговий пакет, ми можемо швидко подивитися в його лічильник, і якщо він наш і коректний - тоді вже перевірити підпис, і якщо підпис теж коректний — розшифровувати дані і передавати їх далі.

Відстеження незмінних даних

Очевидно, що шифрування даних - процедура оборотна, їх можна розшифрувати, а відтак, одні і ті ж дані, зашифровані з одним і тим же ключем, завжди виглядають однаково. Отримуючи пакети від кінцевого пристрою, у якого не змінюються дані, є можливість, не розшифровуючи пакет, зрозуміти, що вони не змінюються.

Протидія таким проявам не складна — повинні змінюватися або дані, або ключ.

Зловмисник дізнався ключ шифрування

Ц випадку, якщо всі пристрої мають один і той же ключ шифрування, власник будь — якого з них може слухати трафік будь-якого іншого пристрою

У LoRaWAN реалізовані дві схеми використання ключів, індивідуальних для кожного пристрою:

- Over The Air Activation, OTAA - ключі генеруються сервером мережі кожен раз, коли пристрій в ній реєструється.

- Activation By Personalization - ключі задані виробником і зберігаються на пристрої, ніколи не змінюючись

Всього ключів використовується як мінімум два - AppSKey, яким шифруються призначені для користувача дані, і NwkSKey, яким підписується повідомлення.

Очевидно, схема з OTAA більш зручна і надійна - ключі можуть змінюватися з тією частотою, з якою хочеться, вони гарантовано унікальні і не відомі нікому, крім мережевого сервера. У ABP ключі не змінюються ніколи, унікальність залежить від сумлінності виробника пристрою (наприклад, ми генеруємо ці ключі з унікального ID мікроконтролера, тому ймовірність їх збігу на двох пристроях незначна), і їх треба десь зберігати в явному вигляді, щоб при підключенні пристрою до мережі прописати їх на сервері.

Перехоплення згенерованих ключів

Очевидно, якщо при реєстрації в мережі кожен раз генеруються нові ключі, то їх треба синхронізувати між пристроєм і сервером, а від так, зловмисник може перехопити їх.

Тому у пристроїв LoRaWAN використовується третій ключ - AppKey, зашитий в пристрій і використовуваний в один момент — при реєстрації в мережі. З його допомогою підписується обмін сесійними ключами між пристроєм і сервером.

В ідеалі AppKey повинен бути унікальний для кожного пристрою, але в багатьох випадках допускається використання одного і того ж AppKey - так як потрібен він всього один раз, що може бути визнано допустимим.

AppKey перед підключенням пристрою заноситься в налаштування на сервері мережі.

Отже, пристрій формує запит на реєстрацію (JoinRequest), не шифруючи його, але підписуючи його ключем AppKey. Сервер мережі, отримавши цей пакет і

перевіривши адреса відправника, підпис, відповідає пакетом JoinAccept, в якому передає налаштування мережі і підтверджує реєстрацію.

Генерація ключів AppSKey і NwkSKey.

Це - результат шифрування AES-128 з ключем AppKey комбінації з переданого сервером у відповіді випадкового числа AppNonce, номера ключа (1 або 2), ID мережі і ще одного випадкового числа DevNonce:

NwkSKey = aes128_encrypt (AppKey, 0x01 | AppNonce | NetID | DevNonce)

AppSKey = aes128_encrypt (AppKey, 0x02 | AppNonce | NetID | DevNonce)

Так як і пристрій, і сервер після обміну пакетами реєстрації знають всі ці параметри, то вони згенерують однакові ключі. Таким чином, ні в який момент ніякі ключі по радіо передаватися самі по собі не будуть, але при цьому пристрій, і сервер отримають унікальні ключі шифрування і підпису пакетів.

Перехоплення потоку даних на себе

У разі, відсилання на сервер перехопленого раніше пакету JoinRequest без його корегування або зміни, сервер відповість на нього пакетом JoinAccept, згенерувавши нові ключі. Після цього пристрій що фактично атакується, просто перестане спілкуватися з сервером, через те, що кінцевий пристрій JoinRequest не ініціював і ніяких підстав оновлювати ключі не бачить. Тобто, та ж атака повтором - але вже на процедуру реєстрації в мережі.

Зловмисник не зможе підробити дані, так як для цього потрібно знати ключі, а для їх отримання потрібно знати AppKey, який він не знає. Але вибити пристрій з мережі - зможе.

Щоб уникнути цього, при реєстрації пристрій передає на сервер випадкове число DevNonce. Крім того, що на його базі генеруються ключі, він служить ще одній меті - сервер LoRaWAN зберігає архів DevNonce. Якщо від пристрою прийшов повторний запит реєстрації з уже використаним DevNonce, сервер його просто проігнорує.

У свою чергу, пристрій повинен кожного разу при входженні генерувати новий DevNonce.

Висновок по першому розділу

На основі документації по протоколу LoRaWAN було проаналізовано архітектуру, принцип побудови і взаємодії пристроїв в мережі LoRaWAN.

В розділі основна увага була звернута на захисні механізми що були вбудовані в протокол, механізми ініціалізації кінцевих пристроїв, взаємодії AS, NS серверів та шлюзів. Процес передачі та шифрування інформації на шляху від кінцевого пристрою до мережевого серверу та серверу додатків, візуалізації.

Крім зазначеного вище в розділі були розкриті деякі питання щодо забезпечення безпеки у відповідності до ймовірних атак.

2. ЗАГАЛЬНА МОДЕЛЬ РОЗРОБКИ ТА ПОСТАЧАННЯ ІОТ ПРИСТРОЇВ

Неоднорідний характер екосистеми IoT та велика вірогідність використання в критичних процесах для організації чи підприємства роблять безпеку та захист пристроїв та інфраструктури ІОТ надзвичайно важливим аспектом, про який потрібно дбати. Хоча більшість постачальників обладнання та програмного забезпечення як LoRaWAN так і IoT різних стандартів в цілому розглядають та застосовують заходи захисту та безпеки під час проектування та розробки їхньої продукції та послуги, застосування заходів та технологій забезпечення безпеки потрібно використовувати на всіх етапах життєвого циклу IoT пристроїв. Окремо слід звернути увагу на те, що без розуміння ланцюгу життєвого циклу ІОТ пристроїв неможливо обрати, або застосувати дієві методи та технології захисту.

Будь-яка організація, яка займається використанням, виробництвом або розповсюдженням фізичних пристроїв та товарів, розуміє поняття ланцюга постачання. Зазначений ланцюг трансформує природні ресурси, сировину, програмне забезпечення та комплектуючи у готовий продукт або послугу IoT, що надається кінцевому споживачу. Такі поняття, як оптимізація та управління ризиками в ланцюгу постачання повинні враховуватися при розробці рішень IoT.

Безпека в Інтернеті речей повинна розглядатися на всіх етапах ланцюга постачання, від раннього концептуального проекту до постачання та обслуговування кінцевого споживача.

2.1 Опис ланцюга розробки та постачання пристроїв ІОТ

Етап проектування пристрою

Перший етап включає створення необхідних проектних ресурсів як для обладнання, так і для програмних компонентів, перш ніж переходити до етапів виготовлення та розробки. Цей процес повинен враховувати особливості ізоляції процесів розробки, тестування тощо.

Дизайн продуктів IoT, як правило, ускладнений через характер побудови, а точніше через складність зв'язку між апаратним та програмним забезпеченням продукту, який як правило, має обмеження (наприклад, вартість, розмір) і базується на апаратному забезпеченні платформи, спеціально пристосовані до сценарію роботи. Завдання проектування в області програмного забезпечення включають, наприклад, збір функціональних та нефункціональні вимоги, підбір початкових версій моделей загроз, проектування архітектур, визначення стека технологій та розробка дрібномасштабних доказів концепцій для оцінки життєздатності. З іншого боку, апаратне забезпечення вимагає створення схем для плат (друкованих плат) та механічні елементи із використанням ECAD (електронний автоматизований дизайн) та MCAD (механічне автоматизоване проектування) інструментів. Потім схеми можна перевірити за допомогою моделювання процеси. Етап проектування продукту є суттєво складним через те, що це змушує дизайнерів продукту розглядати довгострокові перспективи і безліч питань та можливостей які можуть виникнути в майбутньому (наприклад, як буде відбуватися віддалене керування, які протоколи будуть використовуватись, які загрози можуть виникнути).

Етап виготовлення напівпровідників мікропроцесорних елементів

Розвиток у галузі виготовлення напівпровідників відіграв важливу, збільшення можливостей та обчислювальних ресурсів пристроїв з низьким енергоспоживанням та вимогами до малого розміру.

Етап виготовлення включає хімічні процеси, що беруть участь у перетворенні сировини напівпровідникові матеріали в кремнієві пластини, виготовлення масок,

що містять візерунки, які після опромінення ультрафіолетовим світлом буде перенесено в кремнієві пластини.

Два чітких і послідовних кроки можна визначити в цьому процесі:

- FEOL (Front End-Of-Line) відноситься до першої частини, де формуються, та виготовляються окремі електричні компоненти (наприклад, транзистори, конденсатори)
- BEOL (Back End-Of-Line) - це друга частина, де між собою утворюються взаємозв'язки компонентів.

Оскільки дефіцит матеріалів стає все більш актуальною проблемою, конкуренція за доступ до ресурсів посилився, як приклад фактична війна між Intel та AMD.

Додатково слід зазначити, що явний розділ між фазами виготовлення часто розмитий. Такі учасники процесу виготовлення, як ливарні, можуть іноді пропонують послуги, що перевищують очікування, завдяки вищій інтеграції між кроками оптимізації витрат та безпосередньо виробництва. Отже етап виготовлення напівпровідників і етапи виготовлення компонентів, можуть розглядатися як одна одиниця. Не всі проекти IoT вимагають розробки та виготовлення спеціальних IC; багато продуктів може бути на основі готових мікросхем, щоб уникнути високих витрат на виготовлення напівпровідників. На відміну від стадії проектування виробів, стадій виготовлення, як правило має велику кількість загроз. Як приклад, більше важко перевірити дотримання умов договорів про нерозголошення інформації, що в свою чергу може завдати ризики в площині захисту інформації в майбутньому.

Етап виготовлення печатних плат

Виробництво друкованих плат - це один з значущих етапів в проектах виготовлення пристроїв IoT, який, як найчастіше вимагає розробки компактних, захищених від агресивних проявів навколишнього середовища, фактично компоненти встановлюються на свої місця на наступному етапі.

Ризики що можуть виникнути на цьому етапі фактично тотожні попередньому.

Етап компонентного збору та програмування елементів

На цьому етапі електронні компоненти монтуються і припаюються до друкованих плат. Програмні модулі або частини інформації, які є невід'ємною частиною блоків і не є безпосередньо пов'язані з фактичною логікою додатків IoT (розроблені на наступних стадіях) завантажуються та ініціалізуються. Можна виділити два різні типи ініціалізації:

- один тип включає модулі, які однакові для всього діапазону пристроїв (наприклад, завантажувач, прошивка);
- інший включає модулі, які змінюються залежно від пристрою (наприклад, ідентифікатор пристрою).

Завершенням зазначеного етапу є відправлення замовнику інтегрованих у свої фізичні корпуси пристроїв.

Проблеми безпеки на цьому етапі виникають через комбінації одного і того ж програмного забезпечення, що працює з різними конфігурації на різних апаратних платформах. Вплив різного обладнання. Потрібно оцінити вплив різних апаратних платформ на безпеку та надійність програмного забезпечення.

Етап програмування пристрою

Цей етап можна визначити як усі завдання, спрямовані на написання, тестування та розгортання функціонального програмного забезпечення на всіх компонентах пристрою IoT.

Залежно від складності пристрою та кількість різних компонентів, для вирішення яких може знадобитися розробка програмного забезпечення на декількох рівнях: прошивка низького рівня (наприклад, завантажувач), мережеві драйвери, драйвери зв'язку, операційна система, проміжне програмне забезпечення

(наприклад, веб-сервер), графічний інтерфейс користувача. Цей етап може потенційно охоплювати більшу частину життєвого циклу продукту.

Команда розробників зазвичай бере участь у фазі підтримки:

- виправлення виявлених недоліків
- впровадження нові функції
- співпраця з командою технічного обслуговування.

Спектр ризиків в площині захисту пристроїв IOT на цьому етапі може налічувати велику кількість загроз, однак слід зазначити загрози інсайдерських атак а також проблеми в вихідному коді через некомпетентність та помилки.

Етап розробки платформи IoT

Платформа IoT складається з усіх служб, необхідних для роботи та підтримки парку пристроїв IoT, як приклад:

- ідентифікація та авторизація.
- платформи для отримання та обробки телеметрії від кінцевих пристроїв.
- керування та конфігурація кінцевих пристроїв.
- API для отримання даних або подій.
- ПЗ шлюзів для адаптації та конвертації протоколів.

Завдання, що виникають на цьому етапі, зазвичай являють собою поєднання розробки спеціальних проектів з урахуванням конкретного контексту використання та інтеграції сторонніх API та служб— незалежно від того, чи експонується він у хмарі за моделлю SaaS (Програмне забезпечення як послуга) чи встановлений у приватні сервери. Важливо зазначити, що використання сторонніх програмних засобів є значним викликом безпеці та захисту пристроїв на всіх етапах, пов'язаних з розробкою програмного забезпечення.

Етап доставки IOT пристроїв

На цьому етапі відбувається логістичні процеси необхідні для ефективної та безпечної доставки та відстежуючи компонентів та пристроїв IoT.

Забезпечення безпеки та захисту пристроїв IOT на цьому етапі полягає в недопущенні витоку унікальних ідентифікаторів пристроїв третій стороні, саботажу або навмисного виводу з ладу продукції, підміні пристроїв компрометованими або такими в яких було навмисно внесено або змінено алгоритми роботи.

Етап введення в експлуатацію

Етап фактично складається з початкових кроків, які потрібно вжити для того, щоб привести IoT-пристрій у повністю робочий стан на місці замовника. Зазвичай для цього потрібна ініціалізація пристрою, налаштування облікового запису користувача, налаштування мережі, реєстрація хмарних служб та будь-яка подальша спеціальна конфігурація пристрою. Це один з найважливіших етапів, на якому повинні застосовуватися належні методи безпеки, особливо кінцевим користувачем.

Існує багато підходів до процедури надання послуг:

- керований кінцевим користувачем
- керований технічним персоналом (кваліфікований персонал, який налаштовує всі необхідні служби)
- автоматичний (пристрій поставляється повністю або частково попередньо налаштований або відбувається автоматичне віддалене конфігурування при ініціалізації пристрою).

Можна виділити наступні підетапи для захисту IOT пристроїв:

- надання інфраструктури відкритого ключа
- оцінка та сертифікація безпеки

- незалежне оцінювання захисту та забезпечення безпеки роботи IOT пристроїв.

Етап технічної підтримки та обслуговування

Підтримку та технічне обслуговування можна визначити як ряд дій та процесів, які здійснюються протягом життєвого циклу пристрою, щоб уникнути погіршення IoT продукту та забезпечення його роботу згідно з вимогами (функціональними, захисними) у певному проміжку часу (наприклад, помилки програмного забезпечення, уразливості нульового дня).

Процеси підтримки та технічного обслуговування можна класифікувати на дві категорії, віддалені та локальні. Перший використовує мережеву інфраструктуру та такі методи, як безпечне програмне забезпечення або оновлення облікових даних для досягнення своєї мети, і його можна вважати більш економічним. Однак чутливість інформації або інфраструктури іноді може не давати змоги віддаленого конфігурування або підтримкм і потребую фізичної присутності. Крім того прикладом може бути віддалене скасування та оновлення облікових даних пристрою за допомогою каналу зв'язку, вимкненого тими самими обліковими даними (користувачем або певною сутністю).

У випадку запиту з боку користувача призначений технік вирішує проблему, керуючи користувачем у необхідних кроках (наприклад, оновлення програмного забезпечення), або віддаленим підключенням до пристрою. Можливості віддаленої допомоги - і пов'язані з ними наслідки - визначаються можливостями, встановленими на стадії проектування продукту.

Забезпечення належної підтримки пристроїв IoT може допомогти у вирішенні проблем, які можуть виникнути навіть за наявності хорошого дизайну як у коректності виконання певних алгоритмів, так і в питаннях безпеки.

Етап відновлення та перепрофілювання пристрою

Цей етап можна визначити як процедуру, яка виконується після закінчення терміну експлуатації пристрою в певному місці. Залежно від свого стану (або

потреб замовника) пристрій буде знищено, перероблено або перепрофільовано для запуску нового робочого циклу в іншому місці.

Процедура відновлення може включати кілька операцій на двох різних рівнях.

- відновлення за допомогою бекапів та архівів
- видалення даних користувачів, повне стирання та встановлення операційної системи і конфігурування.

В площині захисту пристроїв загроза на даному етапі може полягати в повторному, неавторизованому використанні окремих елементів або елементів пам’яті пристрою для отримання збереженої інформації у разі неповного знищення, або у разі не проведення знищення як такого.

2.2 Опис безпечного життєвого циклу розробки програмного забезпечення (SDLC)

Програмне забезпечення лежить в основі будь-якої системи та послуги IoT, забезпечуючи їх функціональність. Прошивка пристроїв IoT, реалізації протоколів і стеків зв'язку IoT, операційні системи (ОС) для продуктів IoT, інтерфейси програм (API) що підтримують взаємодію та підключення різних служб IoT, архітектура, що покращують взаємодію IoT, драйвери пристроїв IoT, хмарне програмне забезпечення а також програмне забезпечення, що реалізує різні функціональні можливості IoT, - ось деякі приклади того, як програмне забезпечення забезпечує суть IoT.

Програмне забезпечення в екосистемі IoT та його роль як центрального гвинтика у всьому життєвому циклі пристроїв IoT створюють ризики для безпеки. Зловмисники можуть використовувати вразливості програмного забезпечення, щоб поставити під загрозу безпеку систем і служб Інтернету речей та вплинути на належну роботу систем і служб. Вся екосистема IoT, беручи до уваги також

Інтернет та зовнішні фізичні системи, що використовують IoT, повинна враховуватися при розрахунку ризику.

Отже, очевидно, що програмне забезпечення IoT є важливим протягом усього життя систем та послуг IoT, щоб забезпечити стійкі, надійні та безпечні рішення. У цьому відношенні життєвий цикл розробки програмного забезпечення IoT (SDLC) в цілому повинен бути забезпечений, а всі залучені зацікавлені сторони повинні взяти до уваги належні міркування від початку процесу розробки програмного забезпечення до обслуговування та утилізації.

Однак захист IoT, і особливо пристроїв IoT, може виявитися складним завданням для розробників програмного забезпечення, якщо обладнання не має базових можливостей захисту. Наприклад, при впровадженні сильного криптографічного алгоритму в стеку програмного забезпечення саме використання модуля довіреної платформи (TPM) в апаратному забезпеченні гарантує, що приватний ключ не може бути відкритий.

Розробка програмного забезпечення для IoT не може нехтувати основним обладнанням, що, в свою чергу, тягне за собою, що підхід безпеки повинен сприймати його як сукупність, де дизайн апаратного забезпечення впливає на розробку програмного забезпечення. Такі елементи, як "Корінь довіри", є хорошими прикладами того, як програмне та апаратне забезпечення пов'язані та взаємопов'язані, що призводить до спільних міркувань безпеки для протистояння поточним уразливостям IoT, таким як вразливості в стеку зв'язку, похідні від апаратної реалізації.

SDLC - це процес, що складається з різних фаз і має на меті забезпечити ефективні системи відповідно до їх конструкції та функціональних вимог. Існує багато способів досягнення цієї мети. Відповідно, треба розглядаючи безпеку на всіх етапах SDLC IoT та застосовуючи відповідні заходи безпеки до відповідних активів, які можуть зазнати впливу, покращується загальна безпека екосистеми IoT.

Захист процесу SDLC IoT передбачає захист процесу SDLC у всіх елементах екосистеми IoT, а саме в кінцевих пристроях IoT, зв'язку, хмарній, серверній системі та додатках для мобільних пристроїв для управління пристроями. Більше того, слід враховувати всі типи програмного забезпечення, що працює на згаданих вище елементах, включаючи, але не обмежуючись прошивкою кінцевих пристроїв, послугами IoT, реалізацією програмного забезпечення, реалізаціями мережевих протоколів, вихідним кодом API, вихідним кодом шлюзів IoT, програмним забезпеченням, що працює у хмарні тощо.

Стає зрозумілим, що складність та неоднорідність різних елементів IoT та типів програмного забезпечення IoT загострюють питання кібербезпеки, і тому зростає потреба в розробці однорідних, хороших практик захисту SDLC.

2.2.1 Етап проектування вимоги

Вимоги є основою для всього, чого слід дотримуватися в циклі розробки IoT. На цьому етапі визначаються вимоги користувача до бізнесу та функціональних можливостей програмного забезпечення. Ці вимоги відображають цільове використання програмного забезпечення і будуть переведені до специфікацій, які будуть керувати рішеннями щодо проектування, розробки та технічного обслуговування та розгортання на пізньому етапі. Відповідно, важливо розглянути питання безпеки на цьому першому етапі розробки програмного забезпечення.

У цьому відношенні на етапі вимог важливо провести попередню ідентифікацію аспектів безпеки програмного забезпечення з урахуванням користувацьких, ділових, правових, регулятивних та функціональних вимог. Орієнтовні вимоги до безпеки включають політику зміни пароля користувача, необхідність впровадження плану відновлення після збоїв, можливість постійного оновлення тощо.

З метою забезпечення якості вважається гарною практикою періодичний моніторинг та перегляд вимог протягом SDLC. Що забезпечить відповідність безпеки загальним вимогам програмного забезпечення та забезпечить послідовність у розробці. Інженерам безпеки доведеться співпрацювати з інженерами програмного забезпечення та бізнес-аналітиками, щоб гарантувати оптимальне зближення двох векторів.

Слід враховувати ймовірність критичних загроз, рівні довіри до облікових записів користувачів, ймовірність реалізації вектора атаки тощо. Отже, фаза вимог у контексті безпеки дає два результати: набір вимог до безпеки, що залежить від контексту (тип підключення, особливості цільового середовища тощо), а також набір вимог до безпеки, що залежить від пропонованих функціональних можливостей за рішенням (бізнес або випадки використання).

Крім цього, ще одним важливим аспектом, який слід враховувати під час визначення вимог, є фізичні або апаратні вимоги, необхідні для розробки (функціональні вимоги), оскільки програмне та апаратне забезпечення тісно пов'язані. Як частина системи, вимоги до безпеки програмного забезпечення можуть мати певні наслідки при виборі фізичного носія (обладнання), на який слід звернути увагу під час процесу SDLC. Зокрема, вимоги до безпеки на етапі узгодження та проектування вимог до пристрою спричинять необхідність обговорень щодо вибору обладнання у визначенні архітектури на етапі проектування. Наприклад, якщо в якості вимоги обрано реалізацію захищеного механізму завантаження, апаратне забезпечення потребуватиме підтримки такого типу механізму Root-of-Trust (RoT), що потенційно вимагає включення фізичних модулів апаратного захисту (HSM) для управління криптографічними ключами. Таким чином, фізичні вимоги стають таким же аспектом для розгляду, як вимоги до зв'язку, необхідної обробної потужності, місця на жорсткому диску тощо.

Аналіз ризиків, а також використання документів передового досвіду (наприклад, OWASP IoT Top10) та керівних принципів значно допомагають забезпечити розробку програмного забезпечення за допомогою заздалегідь

визначених контрольних списків найпоширеніших ризиків та пасток. Аналіз ризиків також передбачає виявлення активів, які складатимуть програмну систему чи послугу, а також їх взаємодії та зовнішніх залежностей. Це може бути використано для визначення критичності активів, потоків даних та дозволених операцій з даними, що дає значний внесок для підвищення безпеки програмного забезпечення.

При складанні вимог слід враховувати концепцію моделювання загроз. Вона передбачає, що потенційні загрози, такі як структурні вразливості, можуть бути ідентифіковані, перераховані та визначені за пріоритетами. Найчастіше для ідентифікації та класифікації загроз використовується методологія STRIDE (підробка, фальсифікація, розголошення інформації, репутація, відмова в обслуговуванні та підвищення привілеїв). Моделювання загроз починається на фазі вимог з ідентифікації критичних активів і завершується на фазі проектування програмного забезпечення, коли ризики оцінюються та класифікуються, і планується їх пом'якшення, обробка або прийняття.

За допомогою IoT цифровий та фізичний світи більше не стоять окремо один від одного. Природа, різноманітні компоненти та області застосування IoT вводять додаткові параметри в рівняння моделювання загроз. Сюди входить розгляд галузевих загроз, варіанти застосування пристроїв, сумісність із застарілими пристроями.

З огляду на динамічну еволюцію екосистем IoT та притаманну їм адаптацію до мінливого контексту, очевидно, що визначення вимог повинно забезпечити гнучкість протягом усього життя продуктів та послуг IoT. Що стосується захисту IoT SDLC, це означає повторювані, циклічні процеси виявлення вимог безпеки та необхідність розглянути всі можливі сценарії використання системи або послуги IoT.

2.2.2 Етап проектування програмне забезпечення

На етапі проектування програмного забезпечення створюються архітектура пристрою та дизайн IoT — рішення. Цей етап передбачає створення набору документів, що описують, які потреби та функціональні вимоги будуть переведені в специфікації системи та, як буде працювати рішення IoT. Тому важливо переконатися, що технічні характеристики відповідають усім вимогам, визначеним на попередньому етапі. Наприклад, на пристроях IoT специфікації паролів користувачів повинні бути включені як вимога. У свою чергу, на етапі проектування це означатиме впровадження функцій управління паролями користувачів за необхідності (циклічність змін паролю, мінімальна довжина пароля, спеціальні символи тощо).

З точки зору безпеки, застосовується підхід, заснований на оцінці ризику, який визначає відповідні загрози для включення безпеки в розробку програмного забезпечення. Вимоги до безпеки, встановлені на попередньому етапі, переглядаються за допомогою моделювання загроз та аналізу техніки атаки. Моделювання загрози є основою діяльності щодо забезпечення безпеки, що здійснюється під час визначення вимог безпеки, але також може бути додатково вдосконалено під час проектування програмного забезпечення.

Крім того, аналіз атаки повинен враховувати потенційні мотивації, наміри та можливості нападника, а також шляхи атаки, системи, вплив та ймовірність. Як зазначено у розділі вимог, ризики також можуть бути спричинені особливостями використання пристроїв IOT.

Ефективна ідентифікація та пом'якшення загроз безпеці на ранній стадії проектування повинна бути пріоритетними, оскільки пом'якшити їх на пізніх фазах життєвого циклу може бути дуже важко. Більше того, інші положення в розробці програмного забезпечення, такі як корінь довіри, відновлення після збою, та інтеграція механізмів безпеки (FOTA, віддалене управління обліковими даними

тощо) на цьому етапі захищають роботу систем IoT та запобігають дорогим впровадженням у рішення IoT після їх розробки безпекових рішень під час віддаленого супроводу або обслуговування.

Архітектура головним чином зосереджується на все охоплюючих, наскрізних проблемах системи IoT, які переслідують переважно високу масштабованість та інтеграцію різноманітних технологій та систем. Архітектура безпеки для IoT в основному спирається на конфіденційність, цілісність та доступність. Крім того, слід враховувати й інші відповідні аспекти, такі як контроль доступу, конфігурація політики або життєвий цикл безпеки. У цій місії допомагають керівні принципи, безпечні шаблони дизайну тощо. Ці принципи застосовуються до проектування будь — якого рішення IoT, хоча можуть бути відхилення в залежності від конкретних функціональних можливостей або обмежень компонента, що розробляється, або контексту в якому буде працювати. Наприклад, хмарні платформи вимагають ролей адміністрування на основі привілеїв, залежно від випадку використання, пристрої IoT можуть вимагати оновлення та заходів, які направлені на мінімізацію вірогідності збою в роботі.

Через використання датчиків та виконавчих механізмів, які виступають межею між логічним та фізичним світами, аспекти безпеки слід враховувати на етапі проектування. Наприклад, якщо принципи проектування не використовуються, програмне рішення може бути створено без урахування "принципу найменших привілеїв", тому зловмисник може використати цю вразливість, щоб взяти під контроль автоматизований процес і спричинити несправність процесу, що з іншого боку може мати значний вплив на безпеку людей.

2.2.3 Етап розробки та впровадження

На етапі розробки та впровадження специфікації та схеми проектування програмного забезпечення, переносяться у код. Тому те, що визначено на двох попередніх етапах, відіграє вирішальну роль в успішному виконанні процесу розробки.

Основа розробки програмного забезпечення IoT захищена надійним кодом. Код слід створювати, перевіряти, інтегрувати, підтримувати та оновлювати з урахуванням аспектів безпеки. Зменшення ризику, оцінене за допомогою моделі загрози, що проводиться на етапі проектування, має бути реалізовано в кодексі. Враховуючи апаратні та програмні обмеження пристроїв IoT, інтеграція безпеки в код цієї екосистеми створює проблеми для розробників, оскільки вони не можуть створити кодову базу з повноцінною безпекою, як у традиційних IT-системах. Код повинен бути оптимізований, щоб забезпечити врахування кожної інструкції, але одночасно забезпечити безпечну практику розробки (наприклад, SANS Top 25 Помилок програмного забезпечення) та відповідні рішення безпеки для всіх різних елементів, таких як інтерфейс доступу, програми, дані та рівні пристроїв, потрібно врахувати та реалізувати.

У просторі IoT частота випуску продуктів може різнитися. Кодування має підтримувати темп, і з цією метою використання вказівок щодо безпечного коду та стандартів кодування може виявитись надзвичайно корисним для розробників для виявлення відомих вразливих місць, уникнення небезпечних практик кодування та використання вбудованих специфічних функцій безпеки, які певні мови програмування можуть запропонувати. Крім того, існують засоби та методи перевірки якості захисту коду. Захищені фреймворки IoT пропонують розробникам швидкий та ефективний спосіб інтеграції компонентів безпеки, запобігання слабким місцям в площині безпеки та забезпечення дизайну з самого початку розробки.

Для забезпечення безперервних операцій безпечного кодування програмного забезпечення розробка програмного забезпечення повинна супроводжуватися постійною інтеграцією найкращих практик та оцінок безпеки. На цьому етапі слід

послідовно розглядати основні заходи тестування. Таким чином, лише підписана збірка чи версія переходить до наступної фази. Реалістичним способом збереження безпеки в середовищі, яке настільки швидко зростає і так швидко змінюється, є автоматизація. Автоматизовані засоби, такі як інструменти аналізу статичного коду, є корисним доповненням до мануальної перевірки коду, щоб допомогти виявити проблеми безпеки на етапі розробки програмного забезпечення. Методологія статичного тестування безпеки додатків (SAST) дозволяє автоматизувати процес захисту та дозволяє на ранній стадії усунути вразливості рівня додатків. Хоча SAST має місце у випадку безперервної інтеграції (Agile, DevOps, DevSecOps), його можна використати мануально в інших програмних методологіях. І навпаки, інші менш автоматизовані заходи, такі як перегляд коду, середовище побудови та складання, методи захисту від фальсифікацій та управління конфігурацією, доповнюють процес тестування та перевірки, визначений для підвищення захищеності та стійкості.

Звичайною практикою для розробників є використання сторонніх API, фреймворків, бібліотек та інструментів, або програмного забезпечення з відкритим кодом для процесу компіляції та побудови. У світі, де немає необхідності винаходити колесо, цей підхід приносить численні переваги, оскільки дозволяє розробникам зосередитись на особливостях конкретного продукту та скоротити час на розробку. Однак ці сторонні компоненти часто трактуються як чорні ящики і є менш ретельними, ніж внутрішньо розроблені компоненти, отже вони несуть певний ризик. Важливо знати про вразливі місця, які породжуються цими компонентами, і їх слід розглянути та оцінити перед тим, як інтегрувати їх у систему IoT. Наприклад, перед використанням цих зовнішніх компонентів слід перевірити конкретні параметри, такі як активна підтримка компонентів OSS у спільноті, поточне використання цих компонентів на ринку, стан готовності до інтеграції, доступна остання версія, супровідна документація, частота та кількість повідомлених CVE тощо. Також хорошою практикою є використання усталених та захищених бібліотек та фреймворків, щоб кінцевий результат був менш схильним до успадкування вразливостей безпеки компонентів, які він інтегрує.

Розробка IoT вимагає стандартного підходу до розробки безпечних продуктів. При розробці програмного забезпечення важливо організувати процес таким чином, щоб розробники могли працювати над новими версіями, які є менш вразливими або надають кращі послуги. Управління конфігурацією інтегрує процеси, політики та інструменти, щоб зробити програмні системи більш безпечними та гнучкими. У зв'язку з цим, Системи контролю версій (VCS), безпечна можливість завантаження та інші еквівалентні методи стають вартими уваги засобами досягнення цих цілей.

2.2.4 Етап тестування та прийняття

Етап тестування включає всі необхідні кроки, щоб переконатися, що розроблене програмне забезпечення насправді відповідає визначеним вимогам та принципам проектування попередніх етапів. З цієї причини на програмному забезпеченні проводяться різноманітні тести (кожен з яких має різну мету). Ці тести можуть бути автоматизованими, напівавтоматизованими або ручними, і мають на меті перевірити як вихідний код (статичний аналіз), так і запущене програмне забезпечення (динамічний аналіз). Автоматизовані тести можуть значно скоротити необхідний час для їх проведення порівняно з ручними тестами, а також можуть підвищити узгодженість та ефективність завдяки високій масштабованості. І навпаки, вони вносять додатковий ступінь невизначеності на етапі тестування і потребують переробки, щоб бути більш ефективними. Ця невизначеність зумовлена величиною кодової основи, до якої застосовуються тести, та поганою конструкцією для більшого масштабу застосувань, оскільки вони можуть створювати різну кількість помилкових або негативних спрацьовувань порівняно з ручним тестуванням. Тому на етапі тестування важливо оцінити конкретні потреби програмного продукту та встановити найбільш підходящу та найефективнішу стратегію тестування та побудувати відповідне середовище тестування (наприклад, змодельоване або емульоване середовище,

клон, цифровий двійник, набори тестових даних, збір результатів для пост - обробки, розмиття, пентестування, пісочниця тощо).

Що стосується безпеки, тестування на цьому етапі допомагає перевірити правильне та ефективне використання визначених заходів безпеки та засобів контролю, а також виявити та виділити потенційні вразливості та слабкі місця, які вже є в розробленому програмному забезпеченні до інтеграції та розгортання. Зокрема, для етапу тестування IoT SDLC важливо врахувати всі елементи екосистеми IoT, як описано раніше, тобто кінцеві пристрої IoT, прошивку та комунікації. Існує додатковий рівень складності у розробці та побудові відповідної стратегії тестування та середовища тестування для систем та послуг IoT, враховуючи безліч взаємозалежностей численних основних елементів IoT. Отже, важливо ретельно планувати та забезпечувати повну та правильну оцінку всіх інтерфейсів, потоків даних та зовнішніх факторів під час тестування та прийняття.

Важливим аспектом тестування та прийняття є перегляд коду (код ревью). Хоча це можна вважати трудомістким рішенням (особливо, враховуючи велику різноманітність багатьох систем і послуг IoT та часте використання сторонніх бібліотек), цей тип тесту може виявити основні програмні вразливості та слабкі місця, які неможливо виявити інші методи (наприклад, логічні бомби).

Для вичерпного переліку тестів SDLC на основі оцінки ризиків та загроз, та з урахуванням вимог безпеки та специфіки програмного забезпечення слід зробити вибір найбільш підходящого набору тестувань.

2.2.5 Етап розгортання та інтеграції

Фаза розгортання та інтеграції слідує за прийняттям програмного забезпечення, що підлягає успішному тестуванню на попередньому етапі, тобто після його затвердження до випуску. Він передбачає інтеграцію всіх необхідних

елементів програмного забезпечення у виробниче середовище та його розгортання. Розгортання слід ретельно планувати, виконувати та повідомляти всім залученим учасникам (наприклад, кінцевим споживачам, виробничим групам, командам розробників, інтеграторам тощо), щоб забезпечити плавне розгортання в цільовому середовищі. Це особливо складно у сфері IoT, враховуючи багато взаємозалежностей та той факт, що рішення IoT, як правило, застосовуються у широко відкритих середовищах, де адміністратори та команда підтримки можуть не мати повного контролю. Інша особливість розгортання IoT включає неоднорідність середовищ розгортання, наприклад Пристрої IoT для програмного забезпечення, хмарні сервери для внутрішніх служб IoT, шлюзи та мережеві компоненти у випадку реалізації протоколів зв'язку IoT тощо. Також може бути так, що проект програмного забезпечення IoT може вимагати аспектів усіх цих можливих розгортань середовища або цільові середовища. Тому вибір правильної стратегії розгортання має першорядне значення і вимагає зважити такі варіанти, як вплив змін на систему або на кінцевих користувачів, терміни розгортання, варіанти повернення на попередню версію, вимоги до простою тощо.

Важливим кроком у безпечному розгортанні IoT є авторизація активів та користувачів. Для цього корисними є адаптивні інтерфейси адміністрування прав користувачів, аутентифікація пристрою та механізми автентифікації користувачів. Крім того, системи IoT, що дозволяють самостійну реєстрацію, повинні забезпечувати адміністратору засоби перевірки, прийняття або відхилення запиту на реєстрацію.

Крім того, ризикам безпеки, пов'язаним із розгортанням, слід надати належний акцент на етапі розгортання та інтеграції. Мета полягає в тому, щоб мати можливість підтримувати стабільну та безпечну роботу програмного забезпечення під час розгортання, і з цієї причини контролюються відповідні показники, щоб забезпечити "здоров'я" запущеного, "живого" програмного забезпечення. Такі показники можуть включати кількість повідомлень про помилки, кількість виявлених вразливостей та слабких місць, кількість спроб використання та ін.

Важлива частина розгортання включає управління змінами, і у випадку IoT SDLC це в основному стосується оновлення програмного забезпечення. Останні в основному підтримують конфігурацію та управління уразливістю, але можуть також знадобитися з інших причин. Усі виправлення повинні слідувати структурованому підходу до управління змінами, а у випадку безпеки - гарантувати, що будь-які оновлення системи зберігають принаймні той самий рівень безпеки, який був передбачений попереднім рішенням. Що стосується розгортання систем та послуг IoT та оновлень програмного забезпечення, додатковий рівень складності походить від того, як вони взаємодіють в ефірі, через бездротові канали чи використовуючи певні сервери для розповсюдження патчів та оновлень. Очевидно, що всі ці елементи збільшують потенційну поверхню атаки, і тому повинні бути пом'якшені відповідними засобами контролю. Оновлення програмного забезпечення та управління виправленнями також є одними із завдань майбутнього етапу, а саме технічне обслуговування та утилізація.

2.2.6 Етап обслуговування та утилізації

Останній етап IoT SDLC передбачає технічне обслуговування та утилізацію. Важливо не ігнорувати діяльність та завдання, які підпадають під цей етап. Це пов’язано з тим, що програмне забезпечення, розгорнуте у виробництві, потрібно постійно підтримувати, щоб забезпечити доступність та цілісність наданих функціональних можливостей. Поширений та адаптивний характер рішень IoT та той факт, що користувачі можуть привласнити інтелектуальну власність або фізичний пристрій, ще більше посилює потребу в операціях з технічного обслуговування. Крім того, враховуючи той факт, що кінцеві пристрої IoT найчастіше розміщуються в неконтрольованих або критично важливих середовищах і навіть у зовнішніх умовах, планування технічного обслуговування повинно включати відповідні функції (наприклад, послідовні процедури

віддаленого безпечного (ефірного) оновлення, оновлення що передбачає відсутність фізичного доступу, відсутність взаємодії з користувачем, терміни оновлення тощо).

Більше того, з точки зору безпеки, управління інцидентами є постійною діяльністю під час технічного обслуговування. Всі елементи програмного забезпечення, а також усі інші невід'ємні частини рішення IoT потребують постійного моніторингу для забезпечення виявлення та реагування на загрози. Таким чином, є гарною практикою проводити часті оцінки вразливості, тести на проникнення, заходи що до підтримки безпеки та виявлення загроз для запобігання атакам та загрозам для хмари, мережі та кінцевих пристроїв IoT, а також програм, розроблених для таких пристроїв. Обмежені та малопотужні пристрої IoT можуть не створювати та не мати змоги зберігати файли журналів, тому підготовка до звітності дуже важлива. Крім того, планування безперервності обслуговування за допомогою автоматичних резервних копій або резервування допомагає підготуватися до несправностей або збоїв у роботі послуг, спричинених інцидентами.

Інші завдання з технічного обслуговування та утилізації включають управління оновленнями програмного забезпечення, дотримання нормативних вимог (шляхом моніторингу відповідної законодавчої та нормативної бази) та безпечну утилізацію програмного забезпечення та пристроїв. Подібно до попереднього етапу, управління оновленнями програмного забезпечення повинно впоратися з особливостями екосистеми IoT. Додатковим елементом, який слід врахувати, є термін служби пристроїв IoT, який в деяких випадках може бути довгим. У поєднанні з управлінням уразливістю виправлення та оновлення безпеки повинні видаватися своєчасно та надійно.

Крім того, важливо забезпечити, щоб у разі делегування третім сторонам (підрядникам) функцій технічного обслуговування IoT , це відбувалося в доглянутих умовах безпеки, тобто контролі доступу, обробці дозволів, аудиті та підзвітності.

Нарешті, коли програмне забезпечення IoT застаріває (наприклад, коли серія продуктів кінцевих пристроїв IoT виводиться з експлуатації), важливо забезпечити безпечне утилізацію для збереження управління конфіденційністю, забезпечуючи механізми видалення даних. Основним ризиком є зловживання різними типами даних, які використовує програмне забезпечення IoT, і які, ймовірно, кешовані для обробки.

2.2.7 Безпека в SDLC

Розуміння стану кібербезпеки - це перший крок до розробки плану збереження та вдосконалення адекватних заходів безпеки. У цьому відношенні моделі зрілості безпеки (SMM) є дуже корисним інструментом, оскільки вони допомагають організаціям визначати свій рівень безпеки відповідно до вимог, які вони хочуть досягти. Зрілість безпеки оцінює розуміння поточного рівня безпеки, її потреб, переваг та вартості її підтримки. Ця оцінка враховує конкретні загрози нормативним вимогам та вимогам відповідності галузі промисловості організації, унікальні ризики, що існують у навколишньому середовищі, та профіль загроз організації. Існує безліч стандартів, які служать інструментами для оцінки безпеки програмного проекту. Деякі з найбільш широко визнаних галузевих стандартів: Загальні критерії (CC), інтеграція моделі зрілості (CMMI), побудова безпеки в моделі зрілості (BSIMM), безпека промислових систем автоматизації та управління частина 4-1 - безпечна розробка продукту, вимоги життєвого циклу (IEC 62443-4-1), або модель зрілості Open Software Assurance (OpenSAMM), та інші.

Протягом усіх етапів отримання пристроїв — захист треба розглядатися як з точки зору необхідності визначення процедури для здійснення безпечного процесу розробки програмного забезпечення та його управління (управління), так і з боку впровадження та виконання необхідних заходів для забезпечення цього захисту під час різних фази SDLC.

Додатковими до заходів безпеки, під час процесу SDLC є процес документування. Це пов’язано зі складністю рішень IoT, кількістю ресурсів, що беруть участь у процесі розробки, кількістю зовнішніх та внутрішніх компонентів, інтеграцією модулів, конфігураціями, конструкціями, вимогами тощо, хороша

документація та система управління документацією, яка підтримує процес SDLC, щоб зробити його зрозумілим, простежуваним та підлягає моніторингу та аудиту.

2.3 Аналіз можливих загроз під час життєвого циклу пристроїв IOT

У цьому розділі представлена низка найбільш актуальних загроз в контексті циклу життя IoT пристрою. Актуальність була визначена шляхом аналізу статей таких фахівців як: J. Gubbi; G. Suciu; M. Soliman; A. Botta; Z. Zhang та інших. а також проведення інтерв'ю з представника компаній "IOTUkraine" та "DEPS Solutions"

2.3.1 Фізичні атаки

Саботаж

Як приклад — монтажний конвеєр під час виробництва пристрою може надати зловмисникам можливість втручатися та вводити дефекти, які в кінцевому підсумку можуть спричинити проблеми (аж до повного вимкнення та несправності виробу) на пізніх стадіях. Загроза нападу на виробничі процеси є актуальною та тісно пов'язаною загрозою в контексті захисту пристроїв.

Може проявитися на етапі компонентного збору та програмування елементів.

Сірі ринки

Дефектна, викинута або втрачена продукція може потрапити на сірі ринки, що існують поза належними каналами збуту. Це може призвести до непередбачених наслідків та додати численні труднощі до впровадження суворих стандартів безпеки та якості, вводячи на ринок неперевірену та ненадійну продукцію.

Може проявитися на етапах технічної підтримки та обслуговування а також відновлення та перепрофілювання пристрою.

Вибір корпусу не відповідного до рівня загроз

Деякі пристрої вимагають фізичного захисту від втручання залежно від сценарію використання. Вибір матеріалів та способу виготовлення повинен відповідати призначеному використанню виробу. Наприклад, якщо пристрій можна легко розірвати голими руками, немає значення, наскільки гарним є програмне забезпечення інтелектуального замка. Окрім того, що на етапі дизайну треба у відповідний спосіб обрати або спроектувати фізичний корпус, також необхідно врахувати, яі порти мають бути у включеному стані, а які фізично відключені. Наприклад, порт технічного обслуговування, який використовується лише у виробництві, може використовуватися зловмисником у полі. Цей порт слід вимкнути або видалити перед польовою установкою.

Може проявитися на етапах технічної підтримки та обслуговування а також введення в експлуатацію.

2.3.2 Втрата інтелектуальної власності

Крадіжка

Зловмисні партнери або виробники, можуть мати можливість незаконно набувати, експлуатувати, зберігати або розповсюджувати інтелектуальну власність та конфіденційну інформацію (наприклад, проектні документи, вихідний код, облікові дані чи інші секрети). Вони дають змогу дізнатися про можливі вразливості конкретних продуктів IoT і можуть слугувати цінним активом для зловмисників. Ця загроза тісно пов’язана зі стратегією захисту від невизначеності

(тобто досягнення безпеки шляхом забезпечення секретності документації та джерел).

Може проявитися на етапах проектування пристрою, виготовлення напівпровідників та печатних плат

Зворотна інженерія

Наслідки зворотного проектування, можливо, подібні до наслідків викрадення. Основна відмінність полягає в методі, що використовується для отримання конфіденційних активів та частин інформації (наприклад, вихідний код із двійкових файлів, глибоке розуміння апаратних блоків). Вони отримані на основі методу спроб і помилок та ретельного вивчення поведінки кінцевого продукту на етапі використання зловмисниками, які не мають доступу до оригінальних файлів або конструкцій. Цей процес може також призвести до виявлення та випуску у загальнодоступне надбання вразливих місць (як у власних, так і сторонніх компонентах) або вбудованого програмного забезпечення. Важливо зазначити, що зворотна інженерія сама по собі не становить загрози, і її слід розглядати як загрозу лише при зловмисних намірах.

Може проявитися на етапах компонентного збору та програмування елементів, програмування пристрою, розробки платформи IoT, технічної підтримки та обслуговування, відновлення та перепрофілювання пристрою.

Клонування та перевиробництво

Надмірне виробництво - це практика виготовлення товару, конструкторські документи та технічні умови які надав законний власник, поза зазначеної кількості з ціллю розпорядження у корисних цілях надлишком. Ці товари видаються оригінальними, але є небезпечними та представляють загрозу для ланцюга поставок. Завод, фабрика, або недоброзичливий партнер може також клонувати фізичні характеристики, прошивку, програмне забезпечення та конфігурацію безпеки пристрою. Розгорнуті пристрої також можуть бути скомпрометовані, а їх програмне забезпечення може бути клонованим. Клоновані пристрої можуть

дешево продаватися на ринку і можуть містити функціональні модифікації, включаючи бекдори. В якості альтернативи справжньому пристрою, клон може бути замінений оригінальним варіантом під час транспортування або введення в експлуатацію.

Можуть виникнути на етапах компонентного збору та програмування елементів, доставки IOT пристроїв, введення в експлуатацію.

2.3.3 Зловживання або преднамірені дії

Атаки магнітним полем, або електро магнітним випроміненням.

Пристрої, розгорнуті в полі, можуть піддаватися зазначеним атакам що засновані на втручанні на електромагнітному рівні, з ціллю пошкодженні системної пам'яті. Можливі наслідки включають атаку відмови в обслуговуванні або вилучення конфіденційної інформації (наприклад, приватних ключів під час генерації у разі активації за допомогою OTAA).

Може виникнути на етапах компонентного збору та програмування елементів, введення в експлуатацію.

Вставка шкідливого програмного забезпечення

Зловмисникам надається можливість вставити зловмисне програмне забезпечення, основною метою якого є надання незаконного доступу або будь-якої іншої функції, яка суперечить передбачуваному використанню системи. Небезпечні механізми оновлення та некоректні служби оновлення є яскравими прикладами таких можливостей для ін'єкції шкідливого програмного забезпечення.

В цьому контексті особливо уразливі шлюзи IoT, це функціональні пристрої, які зазвичай зустрічаються в архітектурах IoT, але можуть також виконувати

функції джерел загроз. Шлюзи IoT, як правило, виконують допоміжну роль в рамках вимог безпеки, однак вони є способом компрометації пристроїв IoT для зловмисників, забезпечуючи доступ до надійних мереж і спосіб отримання даних з підтримуваних кінцевих пристроїв або терміналів.

Може зустрічатися на етапах компонентного збору та програмування елементів, виготовлення напівпровідників мікропроцесорних елементів, програмування пристрою, розробки платформи IoT, введення в експлуатацію.

Використання інтерфейсів налагодження, або службових інтерфейсів

Налагодження пристроїв IoT без шкоди для конфіденційності, цілісності та доступності є актуальною проблемою - не існує стандартів для включення інтерфейсів налагодження, таких як JTAG. Апаратні або програмні інтерфейси, спеціально призначені для внутрішнього використання в організації, можуть бути неналежним чином вимкнені і потрапити до частини остаточних проектів, що досягають етапів виробництва та складання. Існування цих інтерфейсів зазвичай приписується етапу тестування та дебагу на ранніх етапах, оскільки вони мають слугувати інструментами для налагодження та виявлення помилок, хоча можуть бути випадки, коли ці інтерфейси включаються зі зловмисними намірами.

Ключовим є безпечне використання цієї функціональності та лише для уповноваженого персоналу, що є галузевою проблемою. Вони забезпечують зловмисникам небезпечний рівень доступу до кінцевого продукту.

Може зустрічатися на етапі введення в експлуатацію.

Підробка або контрафакт

Підроблені товари продаються несанкціонованими постачальниками, які не є частиною офіційного каналу збуту виробника. Ці вироби, розроблені та виготовлені невідомими сторонами, позначені як оригінальні вироби певного виробника.

Ця загроза передбачає включення підроблених чіпів до плат, які містять якусь шкідливу модифікацію (наприклад, апаратні трояни) або які не були належним

чином перевірені. Плати, що представляють цю проблему, називаються фальсифікованими платами. Ці несанкціоновані чіпи варіюються від подібних деталей з меншими допусками та можливостями, дефектних деталей, які потрібно було утилізувати, деталей повторно використаних з інших плат, які не відповідають стандартам якості, надмірно вироблених деталей або деталей, вироблених в результаті несанкціонованого використання інтелектуальної власності. Вікно можливостей для втручання може з'являтися на багатьох етапах, включаючи доставку, особливо при роботі з логістичними компаніями, які не мають прозорості щодо своїх заходів безпеки.

Може зустрічатися на етапах виготовлення напівпровідників мікропроцесорних елементів, виготовлення печатних плат, компонентного збору та програмування елементів, доставки IOT пристроїв, відновлення та перепрофілювання пристрою.

2.3.4 Загрози юридичного характеру

Невідповідність стандартам та нормам

Архітектура процесів навколо конфіденційності або шифрування - це проблема, на яку впливають чинні закони та правила щодо конфіденційності, а також той факт, що деякі учасники екосистеми ланцюга поставок мають своє різне розуміння аспектів безпеки, SLA підписуються між різними учасниками ланцюга поставок, щоб забезпечити загальний контрактний вимушений погляд на аспекти безпеки. Усі пристрої повинні відповідати вимогам безпеки, передбаченим відповідними галузями (наприклад, енергетичною, медичною, автомобільною). Більше того, GDPR та будь-які інші місцеві регулювання слід застосовувати для

покриття ризиків, пов'язаних із невідповідністю стандартам, правилам, або законам.

Можуть зустрітися на етапі проектування пристрою, введення в експлуатацію, технічної підтримки та обслуговування.

2.3.5 Загрози псування або втрати інформації

Мережеві загрози

Системи, які необхідні для контролю процесів і існують у мережі, можуть бути скомпрометовані без належних політик безпеки та наявних програмних або апаратно — програмних засобів захисту мережі. Ці активи можуть бути використані для організації, наприклад, широкомасштабних атак відмови в обслуговуванні (DoS) або для погіршення функціонування пристроїв IOT, у разі наявного доступ до Інтернету, вони є найбільш вразливими, хоча окремі внутрішні мережі також мають ризики від інсайдерських атак.

Можуть зустрічатися на етапах проектування пристрою, програмування пристрою, введення в експлуатацію.

Використання заводських налаштувань автентифікації

Пристрої, які вимагають автентифікації, ніколи не повинні виходити з заводу з фіксованими глобальними обліковими даними за замовчуванням або обліковими даними, отриманими з легкодоступної інформації (тобто MAC-адреси). Кожен пристрій повинен мати унікальні випадкові дані, призначені йому під час виготовлення. Особливо під час будь-яких оновлень, які представляють важливу критичну точку безпеки.

Можуть зустрічатися на етапах проектування пристрою, компонентного збору та програмування елементів, програмування пристрою, розробки платформи IoT, введення в експлуатацію, технічної підтримки та обслуговування.

Невизначені програмні чи апаратні збої в роботі пристроїв

Системи, які будь-яким чином пов'язані з функціонуванням, в ідеалі повинні в повному обсязі контролюватися для раннього виявлення збоїв програмного або апаратного забезпечення. Більш активний підхід до виявлення зазвичай призводить до зменшення кількості порушень.

Можуть зустрічатися на різних етапах.

Помилки користувачів

Користувачі повинні бути належним чином поінформовані та підвищувати обізнаність про функціональність та ризики безпеки.

Чи це стосується внутрішніх членів організації, що експлуатують критичні системи та інструменти, або кінцевих споживачів, чиї компрометовані пристрої можуть бути використані для отримання доступу до інших вузлів, які можуть порушити нормальний стан інфраструктури IOT.

Ненавмисні людські помилки можуть бути найпрямішим підходом до проникнення в адекватно захищену систему. Перехоплення повідомлень до інших зацікавлених сторін, пов’язаних із життєвим циклом пристроїв (наприклад, закупівлі), та інші атаки, що виникають внаслідок методів соціальної інженерії, є важливими загрозами, які слід розглядати в контексті помилок користувачів.

Можуть виникати на етапах введення в експлуатацію, технічної підтримки та обслуговування, відновлення та перепрофілювання пристрою.

Технологічні перетворення протягом життєвого циклу пристрою

Технології постійно розвиваються. Ця еволюція може призвести до несподіваних вразливостей, яких не було на етапах проектування та впровадження, це особливо впливає на пристрої з тривалим життєвим циклом. Приклади таких уразливостей включають пристрої, яким не вистачає продуктивності для роботи сучасного шифрування після виявлення дефекту в попередніх схемах або відмови від програмної підтримки з боку постачальників.

Можуть зустрічатися на етапах введення в експлуатацію, технічної підтримки та обслуговування.

Використання неоновлюємих пристроїв та систем

Поширеним явищем є виявлення вразливостей протягом життєвого циклу пристрою, які не враховувались на перших етапах виробництва та використання. Це насправді очікувано, оскільки жодна система ніколи не може вважатися абсолютно безпечною.

Неможливість інтеграції механізму оновлення програмного забезпечення на етапі проектування може становити серйозну загрозу, оскільки позбавляє виробника можливості реагувати на ці проблеми безпеки. Більше того, цей механізм повинен реалізовувати всі технічні заходи, щоб уникнути фальсифікації коду та забезпечити справжність розгорнутого програмного забезпечення.

Може зустрічатися на етапах проектування пристрою, програмування пристрою, технічної підтримки та обслуговування.

Збої в хмарних сервісах

Системи, які залежать від хмарних служб і є критично важливими для функціонування, повинні мати можливість виконувати свої основні функціональні можливості, навіть коли вони відключені на тривалий час. Організації повинні розглянути можливість припинення діяльності постачальника послуг, забезпечивши доступ їх даних у будь-якій формі резервного копіювання. Заходи безпеки для обробки доменних імен, для хмарних служб також повинні бути впроваджені.

Можуть зустрічатися на етапах програмування пристрою, розробки платформи IoT, введення в експлуатацію.

Збій процедур відновлення

Через атаку систему або кінцевий пристрій не вдається відновити, що впливає на функціональність та безпеку. Протягом життєвого циклу пристрою IoT може

знадобитися оновити кілька об’єктів (прошивка, конфігурація, облікові дані). Потрібно враховувати ланцюг довіри, оскільки залежно від того, який актив повинен бути оновлений (зазнає впливу), повинні використовуватися різні механізми. План відновлення повинен визначати, який механізм і який процес слід виконувати, щоб виправити будь-яку потенційну ситуацію, яка може загрожувати службі та безпеці пристрою. Залежно від рівня критичності та порушеного елемента ланцюга, механізми повинні бути тими чи іншими. Це критично важливий процес, при якому може бути порушена безпека пристрою та системи.

Може зустрічатися на етапах введення в експлуатацію, технічної підтримки та обслуговування.

Атака на процедури реєстрації

Відсутність процедур реєстрації або ненадійних механізмів реєстрації може призвести до того, що зловмисники реєструють шахрайські пристрої або перешкоджають реєстрації справжніх пристроїв. Пристрої повинні бути зареєстровані у відповідних службах платформи IoT для аутентифікації після ініціалізації пристрою в продуктовій лінійці та перед наданням кінцевому користувачу, щоб надати йому доступ.

Може зустрічатися на етапі програмування пристрою, розробки платформи IoT,введення в експлуатацію, технічної підтримки та обслуговування.

Використання відновлених або перероблених компонентів

Організації можуть вибрати повторне використання компонентів або деталей, які вже були у використанні, або відновлені, це може бути зроблено з причин оптимізації витрат. Використання компонентів, які вже вийшли з експлуатації та, можливо, не були належним чином перевірені для повторного використання становить загрозу та можуть нести загрозу партії пристроїв.

Може виникати на етапі відновлення та перепрофілювання пристрою.

Атака на виробничі процеси

Виробничі процеси є високочутливими точками входу в життєвий цикл пристроїв IOT. Процеси, що не застосовують адекватних заходів для регулювання та контролю доступу персоналу, можуть спричинити серйозні уразливості; це в свою чергу може призвести до інших обговорюваних загроз, таких як саботаж або ін'єкція шкідливого програмного забезпечення.

Можуть виникати на етапах виготовлення напівпровідників мікропроцесорних елементів, виготовлення печатних плат, компонентного збору та програмування елементів,

Висновки до другого розділу

В першому розділі була проаналізована модель життєвого ланцюга пристроїв IOT, а також проаналізовані та наведені можливі ризики з групуванням по етапам життєвого циклу. Наведені роботи в рамках написання магістерської роботи необхідні для розкриття важливості розуміння повного життєвого циклу пристроїв IOT в контексті захисту кінцевих пристроїв, шлюзів, та інфраструктури в цілому.

Фактично в розділі було доведено необхідність використання захисту пристроїв від різноманітних, як навмисних, так і не навмисних загроз на кожному з етапів життєвого циклу продукту.

3. РЕКОМЕНДАЦІЇ, ПРОЦЕСИ ТА ТЕХНОЛОГІЇ ЗАХИСТУ ПРИСТРОЇВ ІОТ

3.1 Рекомендації що до взаємодії з учасниками створення продукту

Пріоритет роботи з постачальниками, які надають гарантії безпеки

У роботі із зовнішніми постачальниками є загроза, пов’язана з відсутністю контролю за їхніми заходами безпеки, однак, це реальність якої не уникнути. Цю загрозу можна мінімізувати, надаючи перевагу компаніям, які впроваджують стандарти, такі як ISO 27036 та ISO 28000. Компанія, яка вимагає схвалення сертифікації, як правило, є ознакою того, що вона готові серйозно працювати над покращенням безпеки під час взаємодії з партнерами. Сертифікація, як правило, є дорогим процесом, який підходить не для всіх організацій. Організації які не стандартизовані, але мають комплексні заходи безпеки та мають прозорість щодо них (наприклад, право на аудит, контрактні вимоги до безпеки), також слід вважати надійними.

Покращення прозорості

Прозорість має вирішальне значення для контролю безпеки під час взаємодії. Зацікавлені сторони, особливо постачальники, повинні бути прозорими, пропонуючи чітку та детальну інформацію про діяльність під час поставок продукції і передачі всієї відповідної інформації на наступниі етап в процесі створення продукту. Підвищений рівень прозорості мав би бажаним побічним ефектом зміцнення довіри між учасниками певних договорних відносин.

Розробка моделей довіри

Довіра між зацікавленими сторонами є однією з найактуальніших і найважливіших проблем, які слід врахувати для забезпечення співпраці по створенню пристроїв IoT (наприклад, як оцінити безпеку двійкових файлів ODM (Original Design Manufacturers) без вихідного коду). Кожна зацікавлена сторона повинна встановити мінімальний рівень довіри відповідно до своїх потреб та досвіду, аналізуючи потік даних та гарантуючи безпеку та конфіденційність у своїх послугах. Моделі довіри визначають основу для надання офіційних гарантій поведінки різних сторін та підвищення безпеки. Слід зазначити, що не існує єдиного підходу до створення моделі довіри. Підхід, заснований на послідовній оцінці ризиків, дозволить організаціям оцінити вплив на бізнес, застосовуючи належні технічні заходи та контрактні зобов'язання (наприклад, аудит).

Погляд на безпеку як безперервний процес

Безпека не повинна характеризуватися як випадкова діяльність або стан. Концепція процесу передбачає потік та офіційний консенсус між зацікавленими сторонами, а також схвалення та прийняття. Безпека повинна бути включена на всі стадії життєвого циклу по створенню пристроїв як безперервний та повторюваний процес.

Тренінги і навчайте співробітників та команд

Як і у багатьох технологічних галузях, домен IoT демонструє швидкі темпи змін. Підтримка кваліфікованої робочої сили, яка має доступ до регулярних тренінгів з питань безпеки та необхідних ресурсів, щоб бути в курсі справ, має велике значення для вирішення проблем безпеки, що виникають. У більшості організацій повинні бути присутніми професійні команди з кібербезпеки та захисту інформації. Ті, кому бракує ресурсів для утримання таких команд, повинні принаймні забезпечити, щоб інші технічні групи мали відповідний рівень знань з питань безпеки.

Зменшення ризиків при розподілі ресурсів

Розробники програмного забезпечення іноді схильні інвестувати значні ресурси для досягнення розширених функціональних можливостей кінцевого продукту, що може мати небажаний побічний ефект від використання зазначених ресурсів із завдань, пов'язаних із безпекою. Це питання може загостритися деякими рішеннями управлінських рівнів, якщо вони відірвані від фокусу розвитку, прекрасним прикладом цього є нереальні терміни. Сприяння процесу розвитку, який враховує ризики при розподілі ресурсів і гарантує, що безпека отримує належну увагу, може мати значний вплив на безпеку проекту в цілому.

Інформування користувачів відносно безпеки та захисту інформації

Значний відсоток користувачів не має знань про конфігурацію безпеки і не до кінця усвідомлює вплив слабкої безпеки. Вразливі пристрої IoT, якими володіють користувачі, іноді можуть використовуватися як точка входу в системи та служби (наприклад, сервери, що використовуються для конфігурування). Тягар безпеки ніколи не повинен залишатися відповідальністю користувача.

Виробник повинен створювати посібник користувача, що містить інструкції щодо безпечного використання його продукції (мається на увазі кібербезпека).

Отримання фідбеку від користувачів відносно безпеки та захисту інформації

Клієнти повинні надавати вичерпну інформацію, що стосується безпеки. Сюди входять, наприклад, можливі вразливості, які можна було виявити під час життєвого циклу продукту або у відношенні програмних оновлень, які розгортаються на пристроях на місцях. Передача цієї інформації є дуже важливим моментом для досягнення постійної безпеки.

3.2 Рекомендації щодо процесів

Прийняття безпеки у процесі проектування

Модулі безпеки слід вважати складовими першочергового значення і враховувати їх у процесі проектування з перших етапів, щоб уникнути загрози, яка виникає, коли модулі безпеки розглядаються як додаткова думка або вважаються менш пріоритетними.

Інтеграція ланцюга довіри повинна бути пріоритетом для забезпечення цілісності апаратних та програмних модулів у пристроях IoT. Використання безпечних методів кодування та тестів, орієнтованих на безпеку (наприклад, тести на проникнення, сканування вразливості), повинно бути включене на відповідних етапах для впровадження та перевірки функцій безпеки. Слід визначити базову лінію безпеки, яка охоплює найважливіші компоненти життєвого циклу IoT пристроїв.

Така модель безпеки повинна охоплювати основні елементи безпеки: захист, виявлення та реагування на аварії. Людські фактори також повинні враховуватися на стадії проектування. Необхідно дотримуватися найкращих практик та дотримуватися їх суворо, щоб уникнути загроз безпеки через неправильні рішення користувачів. Включення юридичних департаментів до оцінок безпеки та конфіденційності - ще одна важлива практика, яку слід інтегрувати у процес проектування. Більше того, експерти з питань безпеки повинні брати безпосередню участь у перших обговореннях концептуального проекту з командою з управління продуктами, щоб вони могли включити свою точку зору при виборі матеріалів відповідно до їхніх засобів безпеки.

Процес збору даних, технології вимірювання та управління даними

Не всі зацікавлені сторони мають ресурси для проведення аудитів безпеки або аналізу, тому більшість виконують припущення щодо довіри в певний момент. Бажано мінімізувати ці припущення, коли це можливо, зберігаючи гарантії конфіденційності для кінцевого користувача. Сучасні інструменти або механізми,

що допомагають у зборі та вимірюванні даних можуть в значній мірі допомогти в цьому плані.

Процес контролю метрик на протязі життєвого циклу пристроїв IOT

Метрики можна створювати та постійно контролювати, щоб забезпечити спостережуваність стану на протязі життєвого циклу пристроїв IOT. Ці показники можуть бути прив’язані до особливостей поточного ланцюжка поставок або мати більш горизонтальний характер.

Метрики можуть розроблятися переважно на початкових етапах проектування та коригуватися ітеративно та безперервно. Приклади таких показників можуть включати розповсюдження версій мікропрограми, які зараз розгорнуті у споживачів.

Процес розробки моделі загрози

Моделі загроз повинні поєднувати поняття як фізичної безпеки, так і цифрової безпеки, що є невід'ємною частиною інформаційних систем. Процес розробки включає розподіл життєвого циклу пристроя на функціональні блоки та перелік активів у цих блоках, щоб пізніше виявити критичні активи та блоки. З цією метою база знань тактики та техніки атаки, або модель загроз, може послужити основою для розробки комбінованих моделей загроз. Сюди також слід віднести, крім загроз, ненавмисні інциденти, які також можуть вплинути на безпеку та продуктивність, що виникають внаслідок помилок в управлінні підвищеною складністю систем, що виникають в результаті використання IoT як в нових, так і в існуючих мережах та системах. Слід застосовувати методологію оцінки ризику, щоб оцінити відносну важливість загроз залежно від критичності домену та здійснити дії для захисту різних стадій, або етапів життєвого циклу. Мотивація кібератаки (наприклад, фінансова вигода, тероризм) також повинні розглядатися для визначення економічно ефективних засобів захисту та контролю за безпекою.

Процес вибору програмного забезпечення

Використання стороннього програмного забезпечення вводить ступінь невизначеності, яка виступає як загроза безпеці. У разі використання певних програмних компонентів, треба щоб вони були задокументовані, включаючи критерії, якими керувались для вибору. Необхідно віддавати перевагу тим постачальникам та ПЗ, що пройшло процес оцінки та сертифікації.

Комплексний аналіз вихідного коду рекомендується для випадків з відкритим кодом, коли неможливо визначити розробників, хто займається підтримкою. Для допомоги в процесі ідентифікації програмного забезпечення організації можуть використовувати програмні засоби, що спеціалізуються на аналізі компонентів, такі як OWASP Dependency-Track, який є інструментом для створення SBOM. Крім цього можна використовувати певне ПЗ для сканування з ціллю виявлення програмних компонентів та вразливостей. Засоби сканування вихідного коду доступні для внутрішніх та відкритих компонентів, тоді як двійкові засоби сканування можуть застосовуватися в контексті закритого джерела. Слід зазначити, що інструменти з відкритим кодом можуть зіграти значну роль у безпеці IoT, оскільки прозорість та відкритість відіграють дуже важливу роль.

Спільнота з відкритим кодом також ефективна при виявленні недоліків та негайному їх виправленні. Галузь отримує велику вигоду, коли виправлення вразливих місць, виявлених у інструментах з відкритим кодом у контексті приватної організації, повертаються до спільноти з відкритим кодом.

Створення плану тестування (тест плану, та іншої документації в рамках забезпечення якості продукту)

Усі рішення Інтернету речей повинні включати комплексний план тестування, щоб переконатися, що продукт має очікувані функції як програмного, так і апаратного забезпечення. Приймальні випробування повинні проводитися незалежно від будь — яких попередніх випробувань, які могли проводитися на більш ранніх етапах. Частина пристроїв повинна бути перевірена в останній

частині виробництва та піддана тестуванню на кібербезпеку для виявлення помилкових конфігурацій або помилок.

Налаштування за замовченням з ціллю забезпечення безпеки

Значний відсоток клієнтів, як правило, ігнорує функції безпеки через зручність або відсутність технічних знань, це як правило, призводить до уразливостей, яких можна уникнути за допомогою відповідного використання функцій безпеки, вже включених до пристроїв та продуктів. Безпека за замовчуванням, повинна бути підходом для виробників та постачальників, тому замовники, яким потрібно відключити безпеку, повинні робити це свідомо та з розумінням можливих наслідків.

Цей підхід базуватиметься на послідовній моделі безпеки, яка обов’язково застосовується, а також забезпечує належний збір, використання, обробку та передачу даних.

Зобов'язання що до забезпечення безпеки на час підтримки

Пристрої IoT, засновані на необслуговуваному програмному забезпеченні, загрожують цілісності та безпеці системи і цілому. Розширена підтримка та своєчасна доставка виправлень безпеки повинні бути враховані в розробці та плануванні IoT продукту. Виробники повинні доставляти виправлення безпеки щонайменше до закінчення гарантійного терміну, а бажано до закінчення терміну підтримки. У будь-якому випадку, період часу, в рамках якого виробник зобов'язується надавати певні захисні виправлення та оновлення, повинен бути чітко зазначений. Такий сервіс повинен бути доступний без додаткових витрат під час гарантійного використання пристрою.

Управління безпекою під час виготовлення

Матеріали та компоненти, виготовлені на перших етапах життєвого циклу пристрою IOT, які не проходять випробування якості або не вважаються готовими

до використання в подальшому виробництві продукту з будь-якої можливої причини, повинні бути знищені та утилізовані безпечним способом.

Це робиться для того, щоб уникнути загрози отримання доступу до компонентів третіми особами, або зловмисниками з ціллю вивчення та виявлення вразливостей або виготовлення підробок шляхом зворотного проектування.

Використання відповідних техніки видалення даних

Зазвичай пристрої відновлюються до заводських налаштувань і очищаються від усіх приватних даних користувачів на етапах виведення з експлуатації та відновлення. Небезпечні методи видалення даних (наприклад, простий процес видалення, який не перезаписує всі сектори вбудованої памяті), можуть залишити сліди приватних даних користувачів, які згодом можуть бути відновлені за допомогою спеціалізованих програмних засобів іншим користувачем, що має доступ до пристрою.

Слід інтегрувати методи безпечного видалення даних, щоб забезпечити ефективне безпечне видалення всіх даних приватних користувачів та даних конфігурації. Деякі з цих методів слід враховувати перед видаленням даних, наприклад, криптографічне стирання, це означає, що цей процес слід застосовувати ще з етапу проектування пристрою.

Процес створення документації та мануалів

Створіть повний набір документації для боротьби з помилками, які включають чіткі вказівки або послідовність дій, які слід застосовувати, як приклад, щодо аспектів управління конфігурацією та відновлення після відмови. Це критичне питання, оскільки відсутність зазначених ресурсів є загрозою безпеки, більше того, наявність заплутаної, або непомірної кількості документації насправді може шкодити.

3.3 Технології захисту пристроїв ІОТ

Технології модернізації та поліпшення пристроїв

Потреба в модернізації та поліпшенні якості та функціональних можливостей пристроїв, як правило, призводить до рішень IoT, де співіснують кілька поколінь пристроїв та програмного забезпечення, які потрібно оновлювати.

Процеси життєвого циклу пристроїв ІОТ повинні бути розширені до кінця терміну експлуатації будь — якого підключеного пристрою, особливо якщо йдеться про оновлення OTA. Оновлення пристроїв IoT фактично складний процес, оскільки продукти, як правило, базуються на різних пакетах з різними платформами та використовують різні інструменти та сторонні компоненти. Планування та управління цими оновленнями є дуже важливим фактором, який слід враховувати.

Технології для контролю та аудиту безпеки

Нові технології можуть допомогти забезпечити захист на етапах життєвого циклу пристроїв ІОТ, і їх слід оцінювати та у разі можливості використовувати.

Приклади таких технологій включають блокчейн, який може бути використаний для забезпечення цілісності в мережах ІОТ. Штучний інтелект, який може допомогти у процесі прийняття рішень з широкого кола питань. Однак необхідно брати до уваги той факт, що штучний інтелект не забезпечує абсолютних гарантій ефективності, і його слід використовувати як додатковий інструмент у значній кількості випадків.

Технології використання механізми для забезпечення внутрішньої валідації

Полягають в інтегруванні апаратних методів, направлених на сокриття реальних конструкцій і алгоритмів в процеси проектування схем, щоб захистити від загроз зворотної інженерії та перевиробництва. Ці методи засновані на

додаванні ключових входів, які не є критичними для фактичної функціональності схеми, але використовуються для перевірки роботи. Виходи зазначених схем не будуть коректними за наявності недійсного ключа та повинні призводити до непрацездатності пристою. Секретні ключі не повинні бути відомими не довіреним партнерам. Встановлення та коректну роботу таких ключів слід активувати пізніше, в процесі кінцевого складання власником розробки, мається на увазі певний виробник. Таким же чином, транспортні ключі або ключі активації слід використовувати для захисту від крадіжок під час транспортування.

Технології захисту прошивок від фальшування

Безпечне завантаження та підписання прошивки - це заходи безпеки, які забезпечують певний рівень захисту від фальсифікацій. Мікропрограми підписується за допомогою приватного ключа, який доступний лише справжньому постачальнику програмного забезпечення, відкритий ключ згодом використовується пристроєм для перевірки цілісності образу мікропрограми. Безпечне завантаження стосується практики криптографічної перевірки всіх програмних компонентів, які беруть участь у процесі завантаження пристрою.

Ці заходи цілісності повинні застосовуватися під час виготовлення пристрою (коли прошивка прошивається при першому завантаженні) і під час технічного обслуговування (OTA) ці криптографічні операції повинні виконуватися разом із захищеним від втручання обладнанням. Ці два заходи можуть бути інтегровані в існуючі угоди про рівень послуг із сторонніми постачальниками. Варто також згадати, що члени GlobalPlatform працюють над розробкою стандартів безпеки, які визначають низку основ безпеки (SRF) (наприклад, корінь довіри, безпечне встановлення мікропрограми) - що може бути використано для забезпечення видимості функцій безпеки в чіпах. Ці заходи цілісності повинні застосовуватися під час виготовлення пристрою (коли прошивка прошивається при першому завантаженні) та під час обслуговування (OTA) криптографічні операції повинні виконуватися разом із захищеним від втручання обладнанням.

Інтеграція IAM систем для пристроїв IOT

Можливість однозначної ідентифікації кожного пристрою IoT має вирішальне значення і має глибокі наслідки, пов'язані з спостережуваністю та підзвітністю. Системи управління особистістю зазвичай включаються в ширший контекст систем управління ідентифікацією та доступом (IAM), які регулюють цикл ідентифікації пристрою та надають послуги автентифікації та авторизації.

Інтеграція апаратного кореню довіри

Корінь довіри - це перший елемент у ланцюзі довіри пристрою, він зазвичай реалізується за допомогою спеціального апаратного компонента, що забезпечує певні криптографічні елементи, які пристрій може вважати надійними. Ці компоненти, як правило, захищені від несанкціонованого доступу на апаратному рівні і можуть бути використані як основа для заходів безпеки, таких як підписання мікропрограми або безпечне завантаження. Існують також програмні альтернативи з меншими витратами, хоча вони значно вразливіші, і як правило, придатні для обмеженого кола програм.

Технології використання дистанційного оновлення

Можливість віддаленого та автоматизованого застосування оновлень для польових пристроїв є критично важливою в процесі забезпечення безпеки. Етапи життєвого циклу більшості пристроїв IoT не є дискретними, тобто подальший розвиток може відбутися після розгортання пристрою, а вразливості що впливають на систему можна виявити пізніше або в результаті фактичної атаки. Здатність швидко реагувати на зміни в навколишньому середовищі та розгортати оновлення для віддалених пристроїв повинна бути включена і врахована на попередніх етапах проектування. Крім того, ці механізми повинні бути надійними для запобігання зловживанню та ін'єкції шкідливих програм.

Технології використання механізмів аутентифікації в схемі

Для підтримки, відстеження та обслуговування пристрою, аутентифікація пристрою є обов’язковою. Фізично неклоніруемие функції (PUF) - примітив, заснований на фізичних характеристиках, що випливають із його процесу виготовлення та забезпечують однозначну ідентифікацію. Це означає, що PUF можна використовувати для визначення справжності даного пристрою, покращуючи відстежуваність пристроїв по всьому ланцюжку поставок. Переваги PUF включають стійкість до інвазивних атак, які вимагають від зловмисника модифікації фізичних характеристик схеми . Слід зазначити, що, крім PUF, для цього можуть бути використані й інші технології, такі як Trusted Execution Environment (TEE) яку виробники кремнію вже можуть включати у свої чіпи.

Висновки до третього розділу

В розділі були проаналізовані та запропоновані моделі взаємодії між учасниками процесу створення, доставки та використання пристроїв IOT.

Проаналізовані процеси та їх тісний взаємозв’язок з площиною захисту та безпеки.

На основі попередніх розділів проаналізовані та наведені дієві технології по захисту пристроїв IOT на різних етапах життєвого циклу, від етапу проектування до етапу утилізації або виведення з експлуатації. Наведено приклади технологій використання як на організаційному, так і на програмному та апаратному рівнях.

ВИСНОВКИ

Під час виконання магістерської роботи було проаналізовано стандарт LoRaWAN, передбачені стандартом способи та підходи що до захисту пристроїв IOT. Були проаналізовані основні етапи життя, та створення ПЗ для пристроїв IOT. Також були розглянуті стандарти та рекомендації по створенню продукту, та забезпечення захисту та безпеки на різних етапах.

Крім цього в роботі було показано необхідність використання підходів та засобів захисту та забезпечення безпеки в процесах від проектування до виведення з експлуатації та знищення пристроїв IOT. Наведені ризики та варіанти їх пом'якшення або обробки.

В результаті проведеного дослідження було з'ясовано наявність великої кількості технологій, процесів а також підходів у напрямку захисту пристроїв IOT, що можуть бути задіяні в залежності від етапу життя пристрою або його розробки, не виконання яких вірогідніше за все призведуть до компрометації, виводу з ладу, підроблення або знищення пристроїв IOT.

Практичне значення роботи полягає у тім, що в роботі зазначені деякі з проблем, які можуть виникнути на етапах життєвого циклу пристроя, а також акцентована увага на необхідності імплементації заходів забезпечення захисту та безпеки на ранніх етапах, які дозволили б суттєво скоротити коло загроз та вірогідність компрометації або будь якого неправомірного зовнішнього впливу. Результати здійснених у магістерській роботі досліджень можуть бути використані при описі, переліку, проблем пов'язаних з сучасним підходом до технологій захисту та забезпечення безпеки пристроїв IOT на різних етапах життєвого циклу.

ПЕРЕЛІК ПОСИЛАНЬ

1. Secure integration of IoT and Cloud Computing/ Christos Stergiou, Kostas E. Psannis, Byung-Gyu Kim, Brij Gupta 2016p.

2. InternetofThings:AsurveyonthesecurityofIoTframeworks/ MahmoudAmmara,GiovanniRussellob,BrunoCrispo 2017p.

3. The Web of Things: interconnecting devices with high usability and performance/ Simon Duquennoy, Jean-Jacques Vandewalle. 2019p.

4. Towards Security as a Service (SecaaS): On the modeling of Security Services for Cloud Computing/ Angelo Furfaro ; Alfredo Garro ; Andrea Tundis 2014p.

5. Buchheit, Marcellus, Mark Hermeling, Frederick Hirsch, Bob Martin, and Simon Rix. 2020. “Software Trustworthiness Best Practices.” An Industrial Internet Consortium White Paper. https://www.iiconsortium.org/pdf/Software_Trustworthiness_Best_Practices_Whitepaper_2020_03_23.pdf .

6. Filkins, Barbara, and Doug Wylie. 2019. “SANS 2019 State of OT/ICS Cybersecurity Survey.” https://radiflow.com/wp-content/uploads/2019/06/Survey_ICS-2019_Radiflow.pdf

7. Cascella, Roberto. 2019. “Challenges of Cybersecurity Certification and Supply Chain Management.” ECSO - EUNITY Workshop.

8. Boyens, Jon M. 2020. “Key Practices in Cyber Supply Chain Risk Management: Observations from Industry.” Preprint. https://doi.org/10.6028/NIST.IR.8276-draft

9. Boyens, Jon M., Celia Paulsen, Nadya Bartol, Kris Winkler, and James Gimbi. 2020. “Case Studies in Cyber Supply Chain Risk Management: Summary of

Findings and Recommendations." NIST CSWP 02042020-1. Gaithersburg, MD: National Institute of Standards and Technology. https://doi.org/10.6028/NIST.CSWP.02042020-1

10. Fagan, Michael, Katerina N Megas, Karen Scarfone, and Matthew Smith. 2020. "Foundational Cybersecurity Activities for IoT Device Manufacturers." NIST IR 8259. Gaithersburg, MD: National Institute of Standards and Technology. https://doi.org/10.6028/NIST.IR.8259

11. Khan, Faud, and David Rogers. 2019. "IoT Cybersecurity Guidelines, Standards and Verification Systems." In . CABA.

12. Emami-Naeini, Pardis, Yuvraj Agarwal, and Lorrie Faith Cranor. 2020. "Specification for an IoT Privacy and Security Label." Carnegie Mellon University.

13. Johnson, Arnold, Kelley Dempsey, Ron Ross, Sarbari Gupta, and Dennis Bailey. 2019. "Guide for Security-Focused Configuration Management of Information Systems." NIST SP 800-128. Gaithersburg, MD: National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-128 .

14. Kyung Lee, Teddy. 2020. "VIA PUF Technology as a Root of Trust in IoT Supply Chain." Global Semiconductor Alliance (blog). 2020.

15. Kissel, Richard, Andrew Regenscheid, Matthew Scholl, and Kevin Stine. 2014. "Guidelines for Media Sanitization." NIST SP 800-88r1. National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-88r1

16. Li, Changting, Zongbin Liu, Lingchen Zhang, Cunqing Ma, and Liang Zheng. 2018. "A PUF and Software Collaborative Key Protection Scheme." In Information and Communications Security, edited by Sihan Qing, Chris Mitchell, Liqun Chen, and Dongmei Liu, Cham: Springer International Publishing.

17. Liao, R, and Z Fan. 2020. “Supply Chains Have Been Upended. Here’s How to Make Them More Resilient’.” In World Economic Forum. Vol. 6. National Cyber Security Centre. 2018. “Supply Chain Security Guidance.” National Cyber Security Centre. https://www.ncsc.gov.uk/collection/supply-chain-security

18. Guin, Ujjwal, Daniel DiMase, and Mohammad Tehranipoor. 2014. “Counterfeit Integrated Circuits: Detection, Avoidance, and the Challenges Ahead.” Journal of Electronic Testing 30 (1): https://doi.org/10.1007/s10836-013-5430-8 .

19. Savage, Warren. 2020. “Design for Security: The Next Frontier of Smart Silicon.” DesignCon 2020.

20. Oliveira, Daniela, Nicholas Wetzel, Max Bucci, Jesus Navarro, Dean Sullivan, and Yier Jin. 2014. “Hardware-Software Collaboration for Secure Coexistence with Kernel Extensions.” SIGAPP Appl. Comput. Rev. 14 (3): https://doi.org/10.1145/2670967.2670969 .

21. Ray, Sandip, Eric Peeters, Mark M. Tehranipoor, and Swarup Bhunia. 2018. “System-on-Chip Platform Security Assurance: Architecture and Validation.” Proceedings of the IEEE 106 (1): https://doi.org/10.1109/JPROC.2017.2714641

22. Ross, Steven J., Ed Gelbstein, Jane Whitgift, Vasant Raval, Rajesh Sharma, Indrajit Atluri, Hemant Patel, et al. 2017. Internet of Things. Vol. 3. ISACA Journal. ISACA. https://www.isaca.org/resources/isaca-journal/issues/2017/volume-3 .

23. Telecommunications Industry Association. 2020. “Securing the Network and Supply Chain with Industry-Driven Standards.” TIA Position Paper. Telecommunications Industry Association.

24. The Linux Foundation. 2019. “Project Alvarium: Enabling Data Confidence Fabrics to Scale Trust Across Heterogeneous Systems.” October 29. https://alvarium.org/

25. The Ponemon Institute. 2019. “Third Party IoT Risk: Companies Don’t Know What They Don’t Know.” https://sharedassessments.org/blog/2019-iotstudy/

26. Trusted Computing Group. 2020. “TCG Guidance for Secure Update of Software and Firmware on Embedded Systems.” Version 1.0 Revision 72.

27. U.S. Department of Defense. 2016. “DoD Policy Recommendations for The Internet of Things (IoT).” United States. Department of Defense. Office of the Chief Information Officer. https://www.hsdl.org/?view&did=799676 .

28. Xu, Xiaolin, Fahim Rahman, Bicky Shakya, Apostol Vassilev, Domenic Forte, and Mark Tehranipoor. 2019. “Electronics Supply Chain Integrity Enabled by Blockchain.” ACM Transactions on Design Automation of Electronic Systems 24 (3) https://doi.org/10.1145/3315571 .

29. Yang, Kun, Domenic Forte, and Mark M. Tehranipoor. 2017. “CDTA: A Comprehensive Solution for Counterfeit Detection, Traceability, and Authentication in the IoT Supply Chain.” ACM Transactions on Design Automation of Electronic Systems 22 (3) https://doi.org/10.1145/3005346

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація)