УДК 004.321

В.І. МАЛІНОВСЬКИЙ, Л.М. КУПЕРШТЕЙН, В.А. КАПЛУН

АНАЛІЗ ОСНОВНИХ ІНФОРМАЦІЙНИХ ЗАГРОЗ І ВПЛИВІВ У СУЧАСНИХ МІКРОКОНТРОЛЕРНИХ СИСТЕМАХ (АНАЛІТИЧНИЙ ОГЛЯД)

Вінницький національний технічний університет,
вул. Хмельницьке шосе, 95, м. Вінниця, Україна, 21021, e-mail: vad.malinovsky@gmail.com

Анотація. В статті розглянуто і приведено матеріали аналізу основних розповсюджених інформаційних впливів і інформаційних загроз у мікроконтролерних системах, які працюють в складі електронних систем сучасних електронних пристроїв і автоматики. Визначено основні розповсюджені інформаційні загрози і шляхи втручання інформаційних впливів. Приведено оцінку їх впливу і коротко розглянуті шліхи їх мінімізації. Основні базові типи розповсюджених кіберзагроз і каналів напряму впливають на стабільність і безпеку роботи самих мікропрограм МК і як наслідок на роботу електронних пристроїв, в яких цей МК входить. Канали надходження інформаційних загроз для мікроконтролерів є їх слабкими місцями, які потребують вивчення з метою подальшого усунення і мінімізації. Це дало змогу оцінити основні уразливі місця в архітектурі МК і в подальшому дозволить спланувати план дії по мінімізації і нейтралізації основних загроз і інформаційних вплив у МК для стабільного і безпечного функціонування електронних систем на базі мікроконтролерів.
Ключові слова: інформаційна загроза, кіберзагроза, інформаційний вплив, вторинний канал, інтерфейс, мікроконтролер (МК), вразливість, мікропрограма, модуль ПЗ, алгоритмічна вітка.
Abstract. The article presents and provides materials for the analysis of the main distributed information. Influences and information threats in microcontroller systems that work as part of electronic systems of modern electronic devices and automation. The main widespread information threats and the ways of intervention of informational influences are determined. An assessment of their impact and short-term ways of minimizing them are provided. The main basic types of widespread cyberthreats and channels directly affect the stability and safety of the microprograms of the MK itself and, as a result, the operation of the electronic devices in which this MK is included. The channels of information threats for microcontrollers are their weak points, which need to be studied in order to further eliminate and minimize them. This made it possible to assess the main vulnerabilities in the MK architecture and, in the future, to plan an action plan to minimize and neutralize the main threats and information influences in the MK for the stable and safe functioning of electronic systems based on microcontrollers.
Keywords: information threat, cyber threat, information influence, secondary channel, interface, microcontroller (MC), vulnerability, firmware, software module, algorithtmic thread.

DOI: 10.31649/1681-7893-2022-44-2-100-113

ВСТУП

Сучасні мікроконтролери та електронні пристрої на їх основі досить активно і стрімко розвиваються і впроваджуються в останні роки та широко використовуються у багатьох сферах діяльності, зокрема в галузі автоматики, електронних систем загального та спеціального призначення, робототехніці, систем управління критичними системами. Останні тенденції демонструють широкий розвиток цифрових та інформаційних технологій мікро контролерів: від впровадження в традиційні сфери побутової електроніки і персональних пристроїв користувачів до систем автоматизації і цифрових технологій високого рівня, пристроїв Інтернету речей (IoT), які впроваджуються у промисловості, сучасних розробках і автоматизації (Industry X.0) . Від безпеки і стабільності роботи МК пристроїв – напряму залежить стабільність і безпека як обчислювального процесу мікропрограми МК, так і результуюча стабільність системи, яка контролюється цим мікроконтроллером. Особливо це стосується критичних систем автоматики і електронних систем із високим ступенем наслідків порушення функціонування [1,2,3], що можуть призвести до значних негативних наслідків і різного роду втрат.

Мікроконтролери також використовуються в сфері прогресивних спеціалізованих систем автоматики, загальної і спеціальної електроніки і критичних систем, в якості основних елементів і модулів керування технологічними процесами. Тому безпека роботи мікропрограм МК – це пряма безпека і стабільність роботи системи автоматики на базі МК [4, 5, 6]. Стабільне, надійне і нормальне функціонування МК без втручань, інформаційних загроз та первинних і вторинних інформаційних впливів – це першочергове завдання, яке повинно бути вирішено для забезпечення роботи електронних систем і систем автоматики, в які цей МК входить. Сучасні МК мають комплексні прогресивні архітектури, різні обчислювальні спеціалізовані блоки, стабільність роботи яких наряду із стабільністю функціонування периферії МК є ключовим фактором роботи систем електроніки і автоматики на базі МК [2, 7, 8].

В останні роки набуває розвитку і швидкого темпу ескалація і підвищення інтенсивності інформаційних впливів та інформаційних загроз на МК у різних точках прояву [1, 2, 3]. Кількість кібезагроз для МК пропорційно збільшується із ростом цифрових технологій в останні роки і особливо у воєнні часи і в роки інформаційних протистоянь [1, 2].

Метою статті є аналіз і оцінювання впливу базових сучасних актуальних кіберзагроз та інформаційних впливів, а також визначення головних векторів інформаційних атак на МК в результаті функціонування мікроконтролера в складі електронних систем і систем автоматизації різних процесів.

1. ПРОБЛЕМА ІНФОРМАЦІЙНОЇ БЕЗПЕКИ СУЧАСНИХ МІКРОПРОЦЕСОРНИХ СИСТЕМ І ПОТЕНЦІЙНІ ШЛЯХИ ЇЇ ВИРІШЕННЯ

Сучасні засоби для інформаційних втручань в промислові МК пристрої дозволяють успішно реалізовувати шкідливий функціонал і втручання в процеси МК шкідливим ПЗ, різними спеціалізованими програмними і апаратними засобами і модулями, а також методами прямого і не прямого доступу [1, 2, 3]. Це дозволяє впроваджувати шкідливий функціонал, шкідливий код: зокрема різноманітні експлойти та шкідливі мікромодулі ПЗ у мікропрограму МК, здійснювати інформаційні впливи на схему МК і його периферію, та безпосередньо втручатись у його архітектуру [5, 6, 7]. В результаті це дозволяє здійснювати втручання і суттєві порушення штатного функціоналу МК та його периферію. По даним аналізу компанії, самі передові позиції в галузі кібербезпеки займають засоби безпеки для інформаційних систем на базі мікроконтролерів, зокрема засоби для аналізу і запобігання інформаційних втручань і кібератак. Розвиток і розробка їх функціоналу і збільшенням широти сфери застосувань, розширення доступних відомих архітектур МК в таких засобах [8] є пріоритетною задачею і значно перевищує рівні розробки сучасних засобів виявлення, попередження і захисту від атак (IPS/IDS/SecD) [1, 2]. І особливо це стосується загроз у формі спеціалізованого вузькоорієнтованого шкідливого ПЗ для МК систем малого обсягу із обсягом коду 0.1-8КБ, яке експлуатує набір Meltdown and Spectre [9 -13] в архітектурі мікроконтролера і направлене на компрометацію систем автоматики, автоматичного управління та промислової електроніки.

З точки зору безпеки мікроконтролери можуть бути класифіковані згідно з цільовими кінцевими додатками:

- Рішення в галузі автентифікації та довірені платформні модулі (trusted platform module: TPM), наприклад, для захисту як самого користувача, так і мереж IoT.
- Банківські та індустріальні ідентифікаційні рішення для класичних компаній-виробників та емітентів смарт-карток на базі МК, що використовуються у сфері обробки платежів, як персональні ідентифікатори, для оплати послуг транспорту та в системах доставки платного контенту для телебачення.
- Мобільні рішення безпеки для рішень на базі SIM-карток у мобільних продуктах та додатках міжмашинної взаємодії – M2M (machine-to-machine).
- Автомобільні рішення для комунікації ближнього поля (NFC, eSE) та систем забезпечення безпечного водіння.

Індустріальні рішення безпеки для безпеки електронних систем та інтерфейсів систем контролю технологічних процесів [1, 2, 4].

2. ОСНОВНІ ІНФОРМАЦІЙНІ ЗАГРОЗИ В МІКРОКОНТРОЛЕРАХ

Проведений аналіз в мікроконтролерних схемах і систем та пристроїв управління [1] показав, що основними факторами інформаційних загроз і втручань є :

- загрози памяті МК;
- загрози інтерфейсів і ліній доступу МК;
- загрози рівня ядра МК.

Проведений аналіз авторами в роботі [1] показав, що точок впливу і здійсненя інформаційних втручань є багато і результуючий вплив від прояву інформаційних загроз може мати досить серьозні наслідки для кінцевого функціоналу роботи МК.

На базі проведеного аналізу і досліджень інформаційних загроз МК було розроблено класифікацію інформаційних загроз для мікроконтролерів (МК), зображену на рисунку 1.

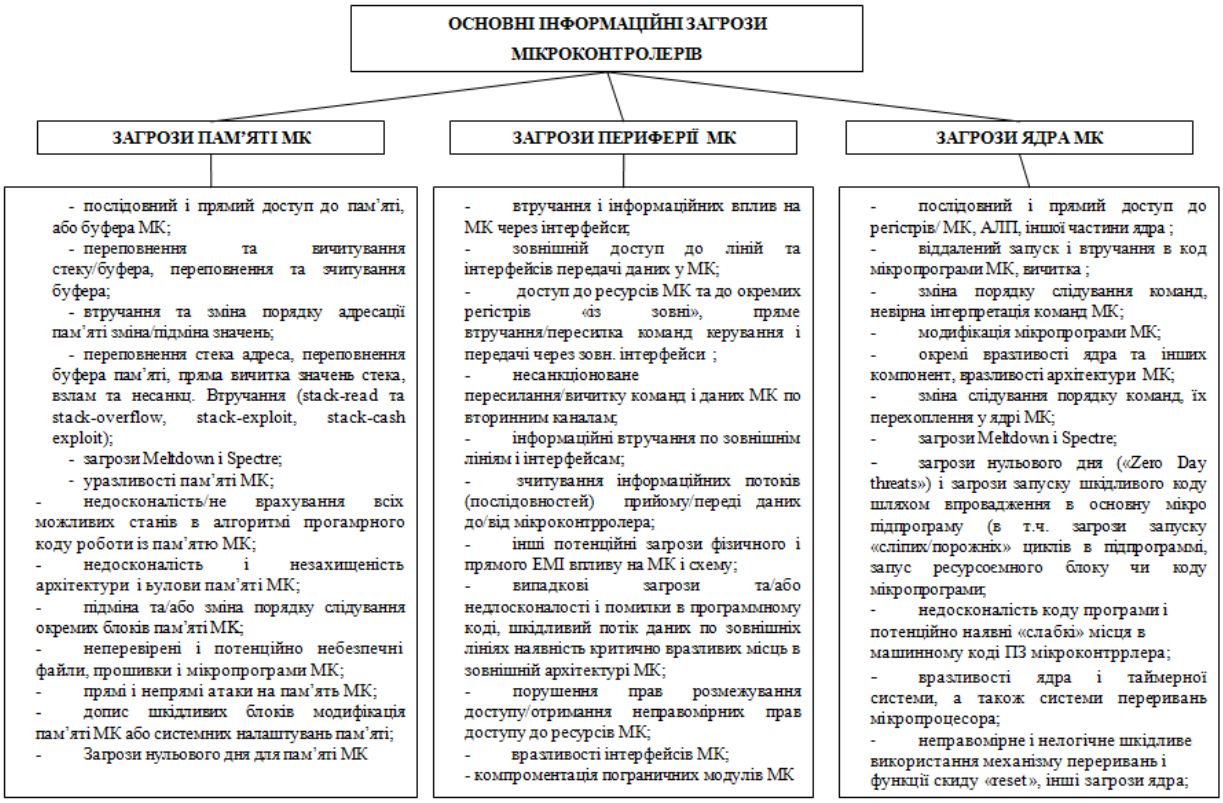

Рисунок 1 – Класифікація основних загроз в архітектурі мікроконтролерів

Враховуючи фактори впливу і їх специфіку прояву загроз для МК [1], пограничні умови і тип будови МК, прояв основних місць прояву загроз показаний на рисунку 2.

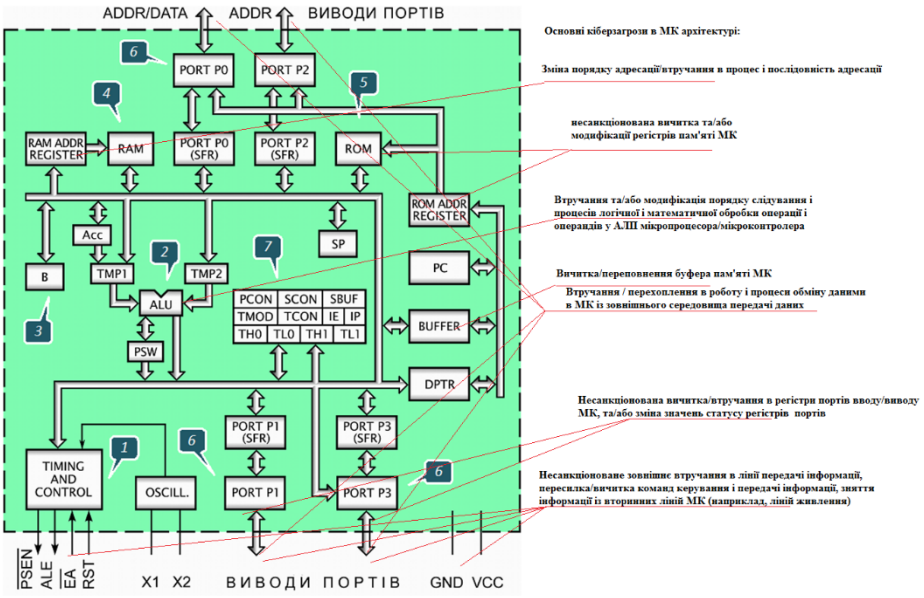

Рисунок 2 – Основні загрози і місця їх прояву в архітектурі мікроконтролера

Рисунок 3 – Зовнішній фигляд фізичного сучасного мікроконтролера і розміщення його в зовнішньому середовищі: на друкованій платі, що передбачає можливість зовнішнього втручання і підключення по стороннім каналам(наприклад, електричним комунікаціям на друковані платі)

На рисунках 2 та 3 показана ілюстрація зон і місця вичитки і запису даних при несанкціонованому втручанні в роботу мікроконтролера із метою впровадження і реалізації різноманітних інформаційних впливів.

3. ОЦІНКА ІНФОРМАЦІЙНИХ ЗАГРОЗ ТА БАЗОВА МАТЕМАТИЧНА МОДЕЛЬ ЗАГРОЗ В МК

Узагальнена математична модель загроз в архітектурах МК і методи мінімізації ризиків була досліджена раніше в окремих роботах по функціонування МК. В роботах [4, 5, 7, 12, 19] наведено огляд моделей безпеки і порушення безпеки в інформаційних системах на базі мікроконтролерів. Дані принципи та інтерпретація процесів частково справедлива і для МК систем із врахуванням окремих особливостей та конкретики і жорсткості опорної архітектури мікроконтролера із врахуванням різних моделей безпеки [4], в яких описані підходи безпеки моделі захисту складних інформаційних систем, в т.ч. IoT. Підходи моделей безпеки [4, 5, 12, 19] використовують імітаційне моделювання для розв'язання проблем поширення інформаційних загроз в середовищах комплексних комунікацій і розподілених блоків і окремих взаємопов'язаних обчислювальних функцій.

Так, модель інформаційної системи на базі МК описується марківськими процесами і функціями складних марківських процесів [21] і графів параметрів із метаданими вкладеності взаємодії інформаційних процесів. Такі графи із Xn – вершинами у вигляді парної взаємодії описуються дискретними функціями і Марківськими процесами оцінки віток обчислювальних процесів у МК.

Спрощено це можна показати як:

$$F(m, n, i, j) = F(X_i, Y_i, n, i) \qquad (1)$$

де $X = \{x_1, x_2, \ldots\ldots, x_n\}$ – множина усіх вершин графу; $E = \{e_1, e_2, \ldots\ldots, e_m\}$ – множина ребер графу.

Кожне ребро метаграфу об'єднує дві підмножини вершин:

$$e_k = (V_i, W_i) \qquad (2)$$

Враховуючи область і вектор взаємодії процесів і роботи сервісів у комунікаційних трактах МК в IoT можна визначити особливості функцій і параметрів для системи характеристик параметрів МК в результаті передачі даних в комунікаціях МК IoT. Враховуючи використання стеку різних спеціалізованих протоколів і інтерфейсів, а також інколи (в окремих випадках) протоколів захисту даних IPSec та PPP які працюють на вищому рівні для захисту каналу і даних в них, можна вважати, що система МК в складі IoT знаходиться під впливом різних факторів і в тому числі факторів інформаційних втручань і факторів загроз кібербезпеки. Основною проблемою в МК є різна обчислювальна і передавальна взаємодія окремих частин блоків команд і спільне використання ресурсів і комунікацій як в архітектурі самого МК, так і в складі електронних схем порівняно із іншими МК та електронними модулями, що входять до цієї схеми. Це робить доступними одні частини ресурсів одних процесів і мікропрограм МК для інших частин мікропрограми і програмних функції [1, 3, 7]. А також інформаційна «прозорість» інформаційно-комунікаційних трактів різних за архітектурою МК пристроїв, що створює можливості для проходження і проникнення інформаційних загроз і впливів із безпосереднім проявом їх у мікропрограмі МК. Крім того, специфіка і особливості використання різних архітектур і високо різність побудови МК не дозволяє використовувати традиційні моделі і засоби безпеки, орієнтовані на мультифункціональні платформи застосовувати їх до МК. Тому вирішення задач інформаційної безпеки

МК вимагає комплексного диференційованого підходу і захисту комунікаційних складових, особливосте архітектур МК і впливів до зовнішніх інформаційних втручань. – як основного вектору атак. Проблемою є також доступність і та безпека зовнішніх комунікацій при дії невизначених процесів і роботою із неперевіреними потоками даних при взаємодії МК із зовнішньою периферією у складі електронних апаратних схем, які також можуть бути скомпрометовані [1, 2, 19].

Рисунок 4 – Принцип і схема розповсюдження інформаційної загрози (на базі експлойту пам’яті із функціями автокопіювання) та доступу до процесів обробки даних у комплексній мікроконтролерній системі із різними взаємопов’язаними МК та іншими елементами

Наявність додаткових комунікацій в інформаційні системі є додатковим фактором загроз і ризиків інформаційної безпеки для МК системи, яка повинні бути враховані в моделях (1) та (2).

Таким чином, досягти максимального рівня захисту в IoT можливо тільки із використанням комплексного підходу використання окремих вищезазначених компонент у вищенаведеній абстрактній формулі захисту.

Забезпечити повну безпеку функціоналу і захищену передачу даних та їх обробку для IoT персонального спрямування із мобільними персональними пристроями користувачів в його складі вкрай складно враховуючи різні функціональне спрямування і використання окремих компонент такої IoT, а також використання каналів Інтернет – як одного із джерела проникнення інформаційних загроз. Забезпечення сталості і надійності функціоналу, концепції цілісності. Доступності та конфіденційності даних (CPA) в таких IoT – є однією із головних завдань. Нові моделі і методи повинні базуватись на комплексному поєднані функціоналу віртуалізації даних, перевірка їх компонентами *IDS/IPS* [1 - 7] у зовнішніх трактах МК в окремих ізольованих програмних контейнерах і інтерфейсах МК для окремих потоків і процесів інформації із змішаним додаткових функціоналом. Також для підвищення рівня безпеки повинні бути створені додаткові умови перевірки і контролю сторонніх інформаційних потоків із надійним вдосконаленим шифруванням із зміщенням та у поєднанні із розпаралелюванням обчислювального процесу із розмежуванням прав доступу на різних рівнях обчислень і віртуальних обчислювальних середовищах(оболонок) для різних процесів.

4. ПРОГРАМНІ І ЛОГІЧНІ ВИДИ І ПРИКЛАДИ ЗАГРОЗ У МІКРОПРОЦЕСОРНИХ СИСТЕМАХ

4.1. Позачергове виконання інструкцій в пам'яті МК - як небезпечний тренд в кібербезпеці ядра в пам’яті МК [7, 8, 9].

У тому чи іншому вигляді позачергове виконання реалізовано у всіх сучасних мікроконтролерах, так як дозволяє отримати найбільший виграш в продуктивності за рахунок паралелізму на рівні команд. Для переналаштування інструкцій застосовується алгоритм Томасуло [10, 11, 12], який дозволяє

уникнути конфліктів і порушення цілісності інформації. Команди впорядковуються в чергу і виконуються в міру готовності своїх операндів. При цьому не обов'язково дотримується порядок їх слідування в мікропрограмі, наприклад [13, 14, 15]:

```
ld → r1, 0(r2) → // завантажити r1 із адреси r2
add → r2, r1, r3 → // r2 := r1 + r3
add → r4, r3, r5 → // r4 := r3 + r5
```

Припустимо, що операнди r3 і r5 готові до використання, в той час як r1 повинен бути завантажений з оперативної пам'яті [16, 17, 18]. При почерговому виконанні інструкцій для виконання першої команди МК зупиняється, чекаючи готовності другого операнда (завантаження з оперативної пам'яті). Але для виконання третьої інструкції дані в r1 не потрібні, а її операнди готові до використання. При позачерговому виконанні ця команда виконалася б 1-ю, в той час поки готується операнд для інструкції завантаження даних з пам'яті. Зауважимо, що позачергове виконання 2-ої команди неможливо, так як вона явно залежить від результатів першої.

Перегрупування інструкцій не повинно порушувати ходу виконання програми, а сам факт позачергового виконання не повинен бути видимий із ззовні. Для усунення помилкових залежностей даних, що не дозволяють перебудувати певні ділянки коду, як метод – використовується перейменування регістрів, а результати позачергового виконання впорядковуються відповідно до мікропрограми.

4.2. Спекулятивне виконання інструкцій і команд в мікропрограмі МК. Мікропрограми в своїй роботі використовують умовні переходи: в залежності від виконання або невиконання певної умови чи інструкції виконуються різні ділянки коду мікропрограми МК. У багатьох ситуаціях перевірка умови - витратна обчислювальна операція (наприклад, вимагає зчитування даних із ОЗП. Позачергове виконання в такому випадку не застосовується, адже невідомо напевно, яка гілка управління повинна бути обрана. Розглянемо приклад:

```
for (i = 0; i < limit; i++)
    results[i] = data;
```

Припустимо, що при компіляції даний приклад зберіг свій вигляд (тобто зберігся цикл). Припустимо також, що було вироблено вже достатньо велике число ітерацій циклу. У такому випадку, якщо перевірка умови циклу займає занадто багато часу, можна зробити припущення: що і в цей і-й раз умова буде виконана, й відповідна операція буде виконана заздалегідь в і+1-й раз. По завершенні перевірки умови стане ясно, чи було припущення коректно. У разі помилки результат доведеться відкинути, але якщо припущення і умова були вірними, то буде зафіксовано результат. Потенційно такий метод дозволив би скоротити час простою процесора і зробити роботу заздалегідь.

Сучасні МК на базі прогресивних архітектур і з широкими наборами інструкцій діють саме таким чином: у МК реалізований предиктор, що робить припущення про те, яка наступна гілка алгоритму мікропрограми МК буде виконуватись в кожен наступний момент часу t_i+1 і може бути задіяний механізм спекулятивного виконання інструкцій, то аналогічно процесу позачергового виконання інструкцій заздалегідь, такі механізми будуть заблоковані і не будуть виконані потенційно небезпечні гілки алгоритму.

Оптимізації, використовувані цими механізмами також призводять до порушення ізоляції процесів і даних в них. І для безпеки такі механізми повинні блокуватись або бути вдосконалені. Для прискорення спекулятивної обробки інструкцій, а також виключення простоїв в очікуванні завершення роботи з оперативною пам'яттю, МК може надати командам, що виконуються спекулятивно, дані із різних внутрішніх буферів: буферів заповнення, буферів зберігання, а також портів завантаження, в яких можуть виявитися необхідні дані. Якщо ці буфери були заздалегідь очищені, то інструкцій можуть бути надані дані інших процесів, до яких не повинно бути доступу. Результати таких операцій будуть відкинуті, але їх можна буде відновити за мікроархітектурними змінами. Крім того, спекулятивне виконання, абсолютно аналогічно позачергового виконання, можуть призвести до витоку даних із кеш-пам'яті, де може розташовуватися інформація з інших процесів і ядра.

4.3. Одночасна багатопоточність. Одночасна багатопоточність – розвиток ідеї і технології суперскалярной архітектури процесора чи багато розрядного мікроконтролера із конвеєрною паралельними або псевдопаралельними блоками обробками даних. Тобто дублювання деяких компонентів і модулів (як програмних, так і апаратних) для виконання декількох завдань одночасно. Це характерно і для мультиядерних архітектур МК. Для системи і МК це виглядає як робота двох різних логічних ядер.

Сучасні мультиядерні процесори і багаторозрядні мультиядерні мікроконтролери (МК) і допускають виконання декількох програм на одному ядрі одночасно або на різних ядрах чи блоках, за

рахунок дублюючих елементів і архітектури конвеєрного типу для процесора чи МК дублювання елементів призводить до порушення ізоляції і загрожує компрометації і конфіденційності даних, зокрема їх витокам. Спільне використання кеш-пам'яті і буферної пам'яті сусідніми потоками у одному ядрі призводить до порушення ізоляції даних і загрожує компрометації і конфіденційності даних, зокрема їъ витокам. Саме спільне використання робить уразливою реалізацію одночасної багато поточності.

4.4. Експлуатація процесорних вразливостей (Meltdown and Spectre) [9, 10, 11]. Експлуатація апаратних недоліків МК, зокрема вразливостей пам'яті, експлуатації мікропрограмних вразливостей вимагає від зловмисника серйозної підготовки для успішного проведення атак. Так, потрібно врахувати всі тонкощі організації архітектури МК, алгоритм роботи атакується системи. такі вразливості аналізуються експертами і заносіться у відповідні таблиці і фіксуються в паспорті конкретну модель МК [12, 13, 14].

У разі атак по побічним каналам, що розглядаються далі, все залежить від виду атаки. Наприклад, атаки через Кеш-пам'ять не вимагають особливо високого рівня підготовки, а лише концептуального розуміння роботи Кеш-пам'ять, і набір мінімально доступних інструментів. У той же час, атаки на основі збоїв й тим більше – атаки аналізу по каналам енергоживлення і енергоспоживання МК вимагають не тільки високих програмних і апаратних навичок і знання системи МК і його схемотехніки, але й розуміння фізичних процесів МК, маніпуляція якими може дозволити витягти або впливати на інформаційні дані в МК.

4.5. Атаки і інформаційні втручання по побічним вторинним каналам. Атаки по побічним (стороннім) каналам - клас атак, орієнтованих на розкриття секретних або закритих даних МК, шляхом непрямого отримання інформації про процеси в МК та їх використання. Для проведення таких атак необхідна наявність побічного каналу в МК, тобто каналу розкриття інформації. Цілями таких атак спочатку були «м'які» впливи, тобто інформаційні впливи на процеси в МК, де неможливо порушити ізоляцію самих алгоритмічних процесів або отримати прямий доступ до секретної чи захищеної області даних в МК безпосередньо. Розглянемо наступний, наочний приклад:

```
if (read_secret() == "secret") {
        quickly_computed_operation(); // Ця операція здійснюється швидко
} else {
        slowly_computed_operation(); // Ця операція здійснюється повільно
}
```

Припустимо, що в коді програми зустрічається наведена вище ділянка. Оскільки одна з гілок алгоритму мікропрограми виконується значно довше іншої, можливо заміривши час роботи області мікропрограми, визначити, була умова виконана чи ні. Таким чином отримується пасивна інформація про виконання процесу. Для даного прикладу можна визначити: чи умова виконується, чи ні. Це аналогічно повному розкриттю секретної та/або захищеної області даних. Якщо час роботи вітки алгоритму мікропрограми МК порівняно невеликий, то ймовірність [3, 4, 5] вичитку секретної або захищеної області даних буде наближатись до 0. Але, якщо цей час є порівняно великим – то ситуація змінюється в протилежний бік, і ризики компрометації захищених даних МК зростають.

Це приклад побічного каналу і вичитки інформації за часом: роль непрямих вторинних ознак отримання інформації виконує сам час роботи окремих гілок алгоритму мікропрограми МК. Перехоплюючи цю інформацію, можна із деякою точністю *G* відновити секретну область даних, не звертаючись до неї безпосередньо.

Атаки по побічним каналам - не рідкість в сучасних умовах і системах МК, і досить часто проявляються на практиці. Наприклад, можна відновити частини використовуваних адрес чи областей даних. Зауважимо, що ці канали існують саме тому, що дані в кеші не розділені між процесами. Це дозволяє спостерігати за змінами у всіх областях даних.

При експлуатації процесорних вразливостей [9, 10, 11], використовується вплив на інформаційний процес у МК, що змушує його розкривати певну область даних за адресою [12, 13, 14]. У багатьох випадках атаки по часу не вимагають фізичного доступу і можуть бути проведені віддалено.

4.6. Атака на основі збоїв. Це різновид активних атак, при яких викликаються спеціальні навмисні помилки в роботі алгоритму мікропрограми, за рахунок чого змінюється вихідне значення. У деяких випадках це дозволяє відновити використовувані секретні захищені дані. В таких атаках найбільш важливо визначити:

- з якою точністю можна вибрати час і місце виникнення помилки;
- яка область даних буде порушена: байт, біт і стек і т.п.;
- параметри помилки: сталість помилки; змінна чи помилка або постійна;
- як себе проявляє збій/помилка: змінюється значення біт з 0 на 1, байт як змінюється і т.п.

Створювати помилки можна різними методами впливу, аж до нагрівання МК і поміщення його в електромагнітне поле і включенням додаткових сигналів в канали передачі та інтерфейси МК. Цікавим є метод варіації і зниження напруги живлення, так як сучасні МК мають функціонал і інтерфейс управління живленням.

Атаки типу **Plunder Volt (вразливість CVE-2019-11157).** Такі атаки використовують побічні канали не за часом, а по помилкам обчислень МК і помилкам в значеннях напруги живлення МК. Як було відмічено раніше, сучасні МК як і процесори (CPU) змінюють робочі частоту і напругу, підлаштовуючи їх під виконуються в поточний момент завдання. Дійсно, якщо використовувати високу частоту і напругу постійно, то витрата енергії *Pw(t)* буде занадто великою. Окрім того, можливі і перегріви кристалу МК. Багаторозрядні МК із АРМ архітектурою сучасного покоління не тільки самі мають механізм зміни і керування частоти і напруги, але й надають інтерфейс управління цими параметрами. Зловживання цим інтерфейсом є основою і можливостями для атаки.

Атаки типу ***Plundervolt*** є досить вимогливими до самих ресурсів атаки, для її проведення потрібні високі привілеї і можливість виконання коду на системі. Крім того, рівень напруги для атаки також визначається зовнішніми умовами, в яких перебуває МК. Проте, це серйозна загроза для апаратних рішень на базі МК, яка вимагає додаткових заходів захисту.

Щоб захиститися від даного виду атак, можна екранувати схему МК із його периферією і ізолювати чіп МК(в т.ч. із встановленням апаратних схемо технічних фільтрів на певні групи частот і інтерфейси), що дозволить уникнути зовнішніх впливів; також можна додати в алгоритм проведення критичною операції перевірки (аж до повторного проведення), що дозволяють оцінити коректність даних і виходу.

4.7. Зчитування залишкової інформації. Зловмисник відновлює збережені в пам'яті дані і отримує з них секрети. Наприклад, може бути відновлений ймовірно віддаленого файлу, цілком або ж частково. У разі використання в МК неочищених мікроструктурних даних із буферу пам’яті можна вичитувати ці дані. Приклад - атака типу «Cold Boot Attack» («холодний запуск» МК), при якій пристрій перезавантажується без завантаження окремих компонент основної мікропрограми. Атаки цього виду можуть бути як віддалені, так і потребують фізичної доступу до пристрою МК. Захиститися від них можливо додаванням етапу очищення залишкової інформації.

4.8. Атаки на стек-пам’яті «Call stack»(Виклик стека) ***і Stack Evaluation*** **(**Еволюція стека)**.** В інформаційних технологіях «***Call stack***» - це втручання в структуру даних стеку пам’яті, що зберігає інформацію про активні підпрограми комп’ютерної програми. Цей вид атаки також відомий як виконання стеку, або переповнення стеку програм, стек керування, стек часу виконання або машинний стек і часто скорочується до «***Call stack»*** **(**"виклик стеку").

Незважаючи на те, що підтримка виклик стека є важливою для нормальної роботи більшості програмного забезпечення, детальне виконання мікропрограми як правило є прихованим та автоматичним на мовах програмування високого рівня. Багато наборів мікропроцесорних інструкцій містять спеціальні інструкції щодо маніпулювання стеками.

*«**Call stack***»* використовується для кількох пов’язаних цілей, але основною причиною їх виникнення є відстеження точки, до якої кожна активна підпрограма повинна повертати контроль, коли вона закінчує виконання. Активна підпрограма – це та, яка була викликана, але ще не завершена, після чого контроль слід повернути до точки виклику. Такі активації стеку підпрограм можуть бути вкладені на будь-який рівень (рекурсивний, як особливий приклади випадок).

Наприклад, якщо мікропідпрограма DrawSquare викликає підпрограму DrawLine з чотирьох різних місць, DrawLine повинен знати, куди повернутися, коли її виконання завершиться. Для цього адреса, що слідує інструкції, яка переходить до DrawLine, адреси повернення, штовхається у верхній частині стека викликів з кожним викликом.

Тому повна структура стека також повинна бути прихованою для сторонніх мікропрограмних модулів. Оскільки загрози типу «***Call stack»*** організовані як правило у вигляді стеку даних програм, абонент висуває адресу зворотного зв'язку в стек, а викликаний підпрограмний режим, коли він закінчує, вичитує адресу зворотного зв'язку зі стеку викликів і передає управління на цю адресу.

Якщо викликана мікропрограма викликає ще одну мікропрограму, вона натискає іншу адресу повернення у стек пам’яті викликів. При цьому інформація накопичується та розкладається відповідно до вимог програми. Якщо показчики стеку пам’яті займають весь простір пам’яті, виділений для стеку пам’яті викликів, виникає помилка, яка називається переповненням стека, як правило, спричиняє збій програми.

У мовах програмування високого рівня (наприклад, C/C++) особливості виклику стека пам’яті і функції виклику стеку пам’яті (Functions of the call stack), як правило, приховані від програміста. Їм надається доступ лише до набору функцій, а не до пам'яті в самому стеку. З іншого боку, більшість мов низького рівня, такі як Асемблер, вимагають залучення навичок програмістів до маніпулювання стеком

прямими та непрямими інструкціями МК. Фактичні особливості функцій роботи із стеку пам'яті мовою програмування залежать від компілятора, операційної системи та доступного набору інструкцій.

Як зазначалося вище, основною метою загроз «***Call stack»*** є зміна адрес і адрес повернення значень даних. Коли викликається підпрограма, потрібно десь зберегти місце (адресу) інструкції, за якою мікропідпрограма виклику може згодом відновитись. Використання стека для збереження адреси повернення має важливі переваги перед альтернативними умовами виклику стеку. Одне з них полягає в тому, що кожне завдання може мати власний стек, і таким чином, підпрограма може бути безпечною для потоків даних,тобто може бути одночасно активною для різних завдань, виконуючи різні дії. Ще одна перевага полягає в тому, що, забезпечуючи повторний вхід, рекурсія віток алгоритму і мікропрограми при цьому автоматично підтримується. Коли функція роботи із стеком пам'яті викликає себе рекурсивно, для кожної активації такої функції потрібно зберігати адресу повернення, щоб потім її можна було використовувати для повернення із активацією цієї функції.

Структура стеку пам'яті в МК із розширеним набором машинних інструкцій забезпечує цю можливість автоматично.

Підпрограмі МК часто потрібен простір пам'яті для зберігання значень локальних змінних, які відомі лише в активній підпрограмі і не зберігають значень після її повернення. Часто зручно виділити місце для цього, просто перемістивши верх стопки на стільки, щоб забезпечити простір. Це дуже швидко, якщо порівнювати з динамічним розподілом пам'яті, який використовує кучу простору. Зверніть увагу, що кожна окрема активація підпрограми отримує свій окремий простір у стеці для локальних ресурсів даних.

Наприклад, виконання вторинних параметрів (*passing parameters)* у функціях роботи із стеком підпрограми часто вимагає, щоб значення параметрів надходили до них за допомогою коду, який їх викликає, і не рідко простір для цих параметрів може бути викладений у стеці викликів. Як правило, якщо є лише кілька вторинних параметрів, для передачі значень будуть використовуватися регістри процесора, але якщо параметрів більше, ніж можна обробити таким чином, знадобиться місце в пам'яті. Стек викликів добре працює як місце для зберігання цих параметрів, тим більше що кожному виклику підпрограми, яка матиме різні значення параметрів, буде надано окремий простір у стеці викликів для цих значень.

Атаки типу є також різновидом атак типу «***Call stack»*** і передбачають, що операнди для арифметичних або логічних операцій найчастіше поміщуються у регістри і там оперують. Однак у деяких ситуаціях операнди можуть бути складені на довільну глибину, що означає, що потрібно використовувати щось більше, ніж регістри. Стек таких операндів, швидше, як у обробнику RPN,називається стеком оцінки і може займати простір у стеку викликів.

Показчик стеку на поточний екземпляр (Stack Pointer) для деяких мов програмування (наприклад, C, C ++) зберігається незмінним із різними аргументами функції виклику стеку під час виклику різних методів в мікропрограмі. Показчик стеку вказує на екземпляр об'єкта в пам'яті, пов'язаний із методом, який потрібно викликати.

Закриття контексту мікропідпрограми (Enclosing subroutine context) дозволяє отримувати доступ до контексту їхніх вкладених підпрограм, тобто параметрів та локальних змінних у межах зовнішньої підпрограми. Таке статичне вкладання може повторюватися – функція, оголошена в межах іншої функції, буде доступна і може бути оголошена в межах первинної функції. Типи атак і їх прояв у стеці пам'яті показаний на рисунку 6.

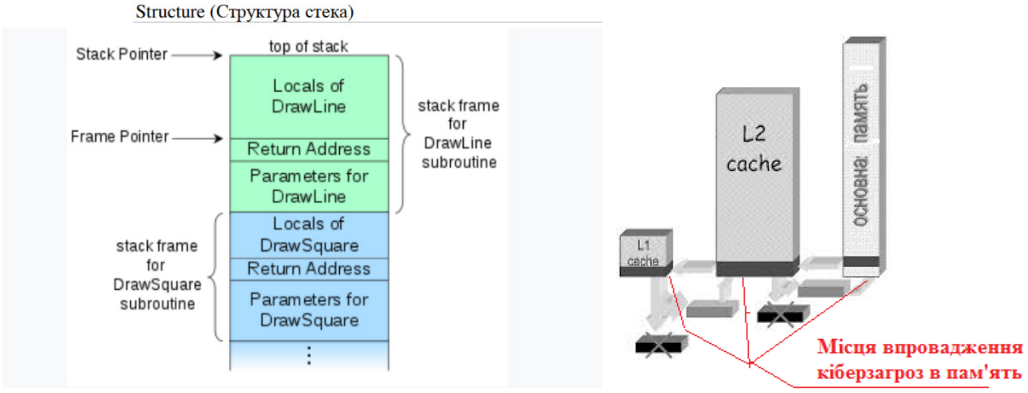

Рисунок 6 - Атаки типу «***Call stack»,*** механізм та місця їх прояву у пам'яті МК (в т.ч. в швикодіючу пам'ять – «кеш-пам'ять" в архітектурі мікроконтролера МК)

Реалізація таких атак (рисунок 6) передбачає засіб, за допомогою якого викликана функція на будь-якому заданому рівні статичного вкладеності може посилатися на огороджувальний кадр на кожному рівні вкладеного вкладання. Зазвичай це посилання реалізовується покажчиком на кадр стеку останнього активованого екземпляра функції, що охоплює, що називається "посилання вниз" або "статичне посилання", щоб відрізнити його від "динамічного посилання", яке посилається на безпосередню адресу (яка не обов'язково повинна бути статичною).

Замість статичного посилання на показчик стеку на вкладені статичні кадри можуть бути зібрані в масив покажчиків дані (рисунок 6), які індексується для пошуку потрібного кадру. Глибина лексичного вкладання є константою, тому розмір є фіксованим.

Макет стека викликів для зростаючого стеку (Call stack layout for upward-growing stacks) Стек викликів складається із сторінок і кадрів стека (їх також називають записами активації або кадрами активації). Це залежні від апаратної організації МК структури даних, що містять інформацію про стан підпрограми. Кожен кадр стека відповідає виклику підпрограми, який ще не завершився поверненням. Наприклад, якщо в даний час працює підпрограма з іменем ProgramDWLine, яка була викликана підпрограмою DwS, верхня частина стека викликів може бути викладена із іншої точки, як на рисунку 6.

Так організовується розміщення даних у стеку викликів. Більше того, незалежно від його структури і архітектури МК, дані і їх структура на сильно відрізняються між собою, і також це показує, що адреси даних або зростають в стеку викликів до вищих адрес або спадають до нижчих адрес, в залежності від структури. Логіка діаграми (рис.) не залежить від вибору адресації. Кадр стека у верхній частині стека відповідає поточній програмі. Макет стека зазвичай включає щонайменше такі елементи (у порядку натискання):

- аргументи (значення параметрів функцій), передані підпрограмі (якщо такі є);
- адреса повернення до абонента підпрограми (наприклад, у кадрі стека DrawLine, адреса в коді DwS);
- місце для локальних змінних підпрограми;
- покажчики стека та рамки **(Stack and frame pointers).**

Коли розміри кадру стека можуть відрізнятися, наприклад, між різними функціями або між викликами певної функції, викидання кадру зі стеку не є фіксованим зменшенням покажчика стека.

При поверненні функції покажчик стека замість цього відновлюється до покажчика кадру – значення покажчика стека безпосередньо перед викликом функції. Кожен кадр стека містить вказівник стека на верхню частину кадру безпосередньо внизу. Покажчик стека - це значення змінного регістру, спільний для всіх викликів.

Інший різновидом впливу «***Call stack»*** *є* повернення іншого стану ***(Other return state).***

Окрім адреси повернення, в деяких середовищах можуть бути інші стани пам'яті МК або мікропрограми, які потрібно відновити, коли підпрограма повертається. Це може включати такі речі, як підвищення рівню привілеїв, інформація про обробку виняткув, невірні арифметичні режими.

4.9. Атаки типу перекриття стеку (Overlap). Для деяких цілей можна вважати, що кадр стека підпрограми та його виклику перекриваються, перекриття, що складається із "суміщених» областей стеку, де параметри і дані передаються від вузла до вузла в точках програми . У деяких середовищах функції мікропрограми записують кожен аргумент у стек, розширюючи таким чином обсяг стеку, а потім викликає окрему область із перевищеного набору. Це призводить до утворення переповнення і перезапису даних у викликаному стеці. В інших випадках функції мікропрограми МК мають попередньо розподілену область пам'яті у верхній частині кадру стека для зберігання аргументів, які він надає в користування також і іншим підпрограмам і функціям в рамках мікропрограммы МК, які він викликає. Цю область іноді називають областю вихідних аргументів або областю додаткової інформації. Згідно з цим підходом, розмір області обчислюється обробником пам'яті МК як найбільший, необхідний для будь-якої званих підпрограм.

4.10. Обробка вхідного виклику (Call site processing). Зазвичай маніпуляція стеком викликів, необхідна на місці виклику підпрограми, є мінімальною (що добре, оскільки для кожної підпрограми може бути багато точок викликів). Значення для фактичних аргументів обчислюються у точці виклику, оскільки вони є специфічними для конкретної точки виклику, або переміщуються в стек, або поміщаються в регістри, як це визначається вживаною умовою викликів.

***4.11. Обробка підпрограмного входу (Subroutine entry processing*).**

У викликаній підпрограмі перший виконуваний код зазвичай називають прологом підпрограми, оскільки він виконує необхідні ведення мікропрограми МК до того, як буде розпочато виконання основного коду для операторів програми. Для архітектур МК із набором команд, в яких інструкції, що використовується для виклику підпрограми, поміщає адресу повернення в регістр, а не надсилає її в стек, пролог зазвичай зберігає адресу повернення, натискаючи значення на стек викликів, хоча якщо

підпрограма не викликає жодних інших процедур, вона може залишити значення в реєстрі. Подібним чином поточні значення покажчика стека або значення покажчика кадру можуть бути змінені.

4.12. Інші види атак. Існують і інші види атак по побічним каналам, розгляд яких виходить за рамки даної статті. Серед них:

- атаки із використанням електромагнітного випромінюванню (аналізується зміна електромагнітного випромінювання в ході роботи);
- акустичні атаки (аналізуються вироблені пристроєм звукові хвилі);
- атаки по випромінюванню (аналізується випромінювання від МК і його інтенсивність);
- атаки зондуванням (вимірювальне обладнання приєднується безпосередньо до контактів МК).

Уразливості мікропроцесорів, викликані наявністю стороннього каналу – найбільший клас вразливостей, їх частка складає понад 90%. Використання побічного каналу спостерігалося у багатьох атаках на МК. Оскільки отримати інформацію з мікроархітектурнимі буферів або ж з реєстрового файлу процесора неможливо, сторонній канал - єдиний засіб отримання захищеної інформації.

У більшості атак на МК використовується побічний канал за часом, а саме – побічний канал через кеш. Для використання цього побічного каналу досить використовувати програмні інструменти, а дороге устаткування і фізичний доступ не потрібні. Крім того, як показано в деяких роботах, використовувати цей побічний канал можна навіть через виконуваний код, що дає зловмиснику ще одну ступінь свободи.

Ряд атак по побічному каналу через кеш і інтерфейс управління пам’яттю здійснюються так: по змінам в кеші визначаються вироблені жертвою дії і, якщо вони визначаються секретними даними, ці дії дадуть атакуючому можливості доступу. Також існують декілька непрямих МК атак, можливих переважно через наявність побічного каналу, це:

***Атаки* TL Bleed.** На даний момент відомо безліч атак на кеші L1, L2 і LLC, і від них були створені захисні механізми, наприклад, Intel CAT. Але, як вказувалося раніше, існують кеші спеціального призначення, зокрема, кеш TLB, який зберігає результати трансляції адрес пам'яті.

***Атаки* TLB** - один з ресурсів, що розділяються потоками на одному ядрі, тому потік може відстежувати щодо змін в даному кеші дії свого сусіда. Для цього змиритися час доступу до певних адресами в пам'яті. Якщо доступ проводиться порівняно швидко, то адреса вже знаходиться в TLB, а значить і сусідить потік його використовував.

Варто зауважити, що реалізація цієї атаки складна, але при цьому її результати неточні. За TLB можна визначити активність за адресою з точністю до сторінки пам'яті (чотири кілобайти в загальному випадку), що далеко не завжди достатньо для визначення дій жертви. Домогтися виконання потоку жертви і потоку атакуючого - непросте завдання, яке, тим не менш, може бути ефективно вирішена за рахунок механізмів операційної системи. Також, потрібно можливість виконувати код на машині жертви і знати відповідність між віртуальними адресами і важливими інструкціями в коді програми жертви.

Така атака може бути ефективна в рамках хмарної інфраструктури, де процеси різних користувачів виконуються на одному МК, і часто використовується багато поточність (технологія : Hyper-Threading).

4.13. *Атаки* за допомогою мережевих інструментів типу MC NetCAT(nc) (вразливість CVE-2019-11184). Дана атака була проаналізована фахівцями Технічного Університету Амстердаму (Vrije Universitet Amsterdam). Уразливість отримала оцінку в 4.8 балів із 10. В основі атаки лежить механізм Intel Data-Direct I/O або DDIO у високо розрядних МК ARM (32-розрядних МК), що дозволяє скоротити час обробки вхідних пакетів за рахунок запису даних безпосередньо в кеш останнього рівня, а не в основну пам'ять. Це прискорює час обробки даних, так як не вимагає тривалої завантаження даних мікроконтролером. З іншого ж боку, це робить кеш-пам'ять доступною для віддаленого зловмисника, причому виконання локального коду експлойту не потрібно. Це експлуатує MC NetCAT: оновлення в кеші відслідковуються віддалено інструментами аналізу потоків даних, що дозволяє провести атаку в рамках мережі, а не тільки одного пристрою. Така схема не вимагає особливих знань про машину жертви, за винятком моделі мікроконтролера.

У атаки є наступні обмеження: по-перше, на МК що атакується, повинені бути включений механізми DDIO; по-друге, із МК має бути встановлено з'єднання RDMA (це механізм прямого віддаленого доступу до пам'яті МК).

Причина першого обмеження очевидна: якщо механізм обміну даними DDIO не включений, провести мережеву атаку не просто. Друге обмеження пов'язане з вимогою до точності вимірів часу і доступу до віддаленої пам'яті. Для успішної атаки зловмисник повинен хоча б частково контролювати те, куди будуть записані його дані, і звідки вони будуть прочитані. Механізм RDMA дозволяє віддаленому влаштуванню безпосередньо перезаписувати або зчитувати заздалегідь схвалені і зареєстровані області пам'яті сервера. RDMA зустрічається в суперкомп'ютерних комплексах і дата-центрах, що звужує сферу застосування атаки.

Автори роботи відзначають можливість використання NetCAT для побудови прихованого каналу повідомлення і для крадіжки даних користувача з SSH сесії. Крадіжці даних сприяє те, що протокол telnet, який використовується в SSH, має на увазі відправку пакета даних на сервер при кожному натисканні клавіші користувачем. Відновивши час приходу пакетів через поновлення в кеші, зловмисник зможе визначити послідовність натискань методами машинного навчання і, в результаті, дізнатися секрети користувача.

Атака або інформаційний вплив на МК є серйозною небезпекою, які можуть призвести до суттєвих наслідків і змін в процесі управління технічною системою із цим МК в складі, до змін в мережі із МК.

Для нейтралізації загроз повинні використовується різні методи і підходи, що базуються на різних принципах і в т.ч. різноманітна програмно-апаратна ізоляція МК і його каналів від потенційних точок впровадження інформаційних загроз. Але для точного виявлення загроз і атак попередньо необхідним є точне визначення і детальний аналіз цих загроз і інформаційних впливів на МК в кожному конкретному випадку і в кожній окремі архітектурі МК із визначенням характеру і специфіки їх прояву. Зокрема комплексні підходи повинні включати максимальну ізоляцію і інформаційний захист системи МК (як програмний, так і апаратний) – ізоляція області пам'яті, шин даних і адрес, для мінімізації втручання в роботу МК, підходи моніторингу, використання криптостійких надійних алгоритмів і попереджувальних засобів фільтрації і блокування інформаційних загроз. Не зайвим, але часто досить затратним по ресурсам є використання платформ і мережевих інструментів і систем аналізу інформаційного трафіку МК. Ефективність цих методів, підходів і систем, та й відповідно досить часто й затрати на їх реалізацію не у повній мірі дозволяють отримати необхідний рівень безпеки і співвідношення вартість / техн. функт. рівень захисту, враховуючи сучасні загрози «0»-го дня і рівень сучасного шпигунського і хакерського програмного забезпечення для МК і систем управління на їх основі.

ВИСНОВКИ

Для нейтралізації і вчасного попередження інформаційних впливів в МК, був проведення аналізу процесів інформаційних загроз у МК з метою вчасного попередження і нейтралізації цих загроз і впливів.

В матеріалах статті було приведено аналіз основних, найбільш актуальних загроз і вразливостей мікроконтролерів, проведено розгляд і наведення основних інформаційних ризиків для МК. Визначено основні вектори і канали надходження інформаційних впливів і загроз в МК та технології здійснення атак. Проведено коротку оцінку їх розвитку. Проведено основні базові типи факторів ризиків і інформаційних впливів на МК, які можуть працювати в складі системи управління Описані і надані базові підходи до захисту від цих загроз і інформаційних втручань в роботу МК. Це дасть змогу оцінити основні уразливі місця в архітектурі МК і уникнути і попередити основні ризики їх прояву.

ПЕРЕЛІК ЛІТЕРАТУРИ

1. В.І. Маліновський. Аналіз загроз безпеки мікроконтролерів / В.І. Маліновський, Л.М. Куперштейн. // Інформаційні технології та компютерна інженерія. - 2022. - Вінниця: УНІВЕРСУМ - Вінниця, ВНТУ – № 3 (55). – С. 21 - 32.
2. В.І. Маліновський. Мінімізація факторів кіберзагроз і спеціалізовані підходи до інформаційного захисту мікропроцесорних систем індустріального Інтернету речей / В.І. Маліновський // Матеріали LI-ої Науково-технічної конференції факультету інформаційних технологій та комп`ютерної інженерії –2022. 31.05.2022. – ВНТУ: [Електронний ресурс]. –: URL: https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2022/paper/view/15000.
3. В.А. Гапанович, Е.Н. Розенберг, И.Б..Шубинский, Некоторые положения отказобезопасности и киберзащищенности систем управления / В.А. Гапанович, Е.Н. Розенберг, И.Б. Шубинский // Надежность. – 2014. – №2. – С.88-100.
4. С.Г. Антонов, С.М. Климов, Методика оценки рисков нарушения устойчивости функционирования программно-аппаратных комплексов в условиях информационно-технических воздействий // Надежность. – 2017. – Том 17. – №1. – С.32-39.
5. С.М. Климов, С.В. Купин, Д.С. Купин, Модели вредоносных программ и отказоустойчивости информационно-телекоммуникационных сетей // Надежность. – 2017. – Том 17, № 4. – С. 36-43. DOI:10.21683/1729-2640-2017-17-4.
6. Cybersecurity and Data Stability Analysis of IoT Devices / Malinovskyi Vadym, Kupershtein Leonid, Lukichov Vitaliy / / Materials of 2022 IEEE 9th International Conference on Problems of Infocommunications. Science and Technology(PIC S&T`2022). - IEEE Ukraine Section. - Kharkiv National University of Radio Electronics
7. Risks Assessment and Approaches to Creative of the Reliable Software Modules for IoT Devices / Malinovskyi Vadym, Kupershtein Leonid, Lukichov Vitaliy . - Materials of International Conference on Innovative Solutions in Software Engineering. - November 29-30, 2022.- Ivano-Frankivsk, Ukraine.

8. Yuan Xiao, Yinqian Zhang, Radu Teodorescu. [Online]. Speech miner: a Framework for investigating and measuring speculative execution vulnerabilities [Електронний ресурс]. – Режим доступу: https://arxiv.org/pdf/1912.00329.pdf
9. Meltdown and Spectre: Which systems are affected by Meltdown?: [Електронний ресурс]. – Режим доступу: https://meltdownattack.com/#faq-systems-meltdown
10. Meltdown and Spectra: Which systems are affected by Meltdown?: [Електронний ресурс]. – Режим доступу: https://meltdownattack.com/#faq-systems-meltdown
11. Speculative Processor Vulnerability [Online]. ARM Developer Forum. Specifications Updated March 8, 2022 [Електронний ресурс]. – Режим доступу: https://developer.arm.com/Arm%20Security%20 Center/Speculative%20Processor%20Vulnerability
12. Cache Speculation Side-channels white paper [Online]. ARM Developer Forum. Specifications Updated March 8, 2022 [Електронний ресурс]. – Режим доступу: https://developer.arm.com/documentation/102816/0205/
13. Kernel Side-Channel Attack using Speculative Store Bypass - CVE-2018-3639 [Електронний ресурс]. – Режим доступу: https://access.redhat.com/security/vulnerabilities/ssbd .
14. ISO/IEC, «Information technology — Security techniques-Information security risk management» ISO/IEC FIDIS 27005:2008.
15. Kakareka, Almantas (2009). 23. У Vacca, John. Computer and Information Security Handbook. Morgan Kaufmann Publications. Elsevier Inc. с. 393. ISBN 978-0-12-374354-1.
16. Serdar Yegulalp Rowhammer hardware bug threatens to smash notebook security / by Serdar Yegulalp// – March 9 – 2015 Електронний ресурс]. – Режим доступу: https://www.infoworld.com/article/2894497/rowhammer-hardware-bug-threatens-to-smash-notebook-security.html
17. Kuljit Bains Patent US 20140059287 A1: Row hammer refresh command, February 27, 2014, by Kuljit Bains et al. Електронний ресурс]. – Режим доступу: https://patents.google.com/patent/US20140059287
18. Cisco Systems security advisory. Row Hammer Privilege Escalation Vulnerability, // Cisco Systems security advisory. - March 11. – 2015 Електронний ресурс]. – Режим доступу: Cisco Systems
19. С.Г. Антонов, С.М. Климов, Методика оценки рисков нарушения устойчивости функционирования программно-аппаратных комплексов в условиях информационно-технических воздействий // Надежность. – 2017. – Том 17. – №1. – С.32-39.
20. Sudhakar Govindavajhala and Andrew W. Appel. Using Memory Errors to Attack a Virtual Machine. // Princeton Edu University press – March 6.–2003. – Електронний ресурс]. – Режим доступу: https://www.cs.princeton.edu/~appel/papers/memerr.pdf.
21. М. В. Карташов. Імовірність, процеси, статистика. – Київ : ВПЦ Київський університет, 2007. – 504 с.
22. Загрози та вразливості бездротових мереж. [Електронний ресурс]. – Режим доступу:http://dspace.kntu.kr.ua/jspui/bitstream/123456789/5022/1/AUConferenceCyberSecuri ty_November2016_p146.pdf
23. Аналіз механізмів захисту та вразливостей бездротових Wi-Fi мереж. [Електронний ресурс]. – Режим доступу: http://ir.nmu.org.ua/butstream/handle
24. M. Swanson. NIST Special Publication 800-34 Rev. 1 Contingency Planning Guide for Federal Information Systems / M. Swanson, P. Bowen, A. W. Phillips, D. Gallup, D. Lynes. – 2010. – 149 p.

REFERENCES

1. V.I. Malinovsky. Analysis of security threats of microcontrollers / V.I. Malinovskyi, L.M. Kuperstein. // Information technologies and computer engineering. - 2022. - Vinnytsia: UNIVERSUM - Vinnytsia, VNTU – No. 3 (55). – P. 21 - 32.
2. V.I. Malinovsky. Minimization of cyber threat factors and specialized approaches to information protection of microprocessor systems of the industrial Internet of Things / V.I. Malinovskyi // Proceedings of the 3rd Scientific and Technical Conference of the Faculty of Information Technologies and Computer Engineering - 2022. 31.05.2022. – VNTU: [Electronic resource]. –: URL: https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2022/paper/view/15000.
3. V.A.. Gapanovych, E.N.. Rozenberg, I.B. Shubynskyi / Some provisions of fail-safe and cyber security of control systems / Reliability. – 2014. – No. 2. - P.88-100.
4. S.G. Antonov, S.M. Klymov. Methodology for assessing the risks of disruption of the stability of the functioning of software and hardware complexes in the conditions of informational and technical effects // Nadezhnost. – 2017. – Volume 17. – No. 1. - P.32-39.
5. S.M., Klimov, S.V. Kupin, D.C. Kupin. Models of malware and fault tolerance information and telecommunication networks // Reliability. – 2017. – Volume 17, No. 4. – P. 36-43. DOI:10.21683/1729-2640-2017-17-4.

6. Cybersecurity and Data Stability Analysis of IoT Devices / Malinovskyi Vadym, Kupershtein Leonid, Lukichov Vitaliy / / Materials of 2022 IEEE 9th International Conference on Problems of Infocommunications. Science and Technology(PIC S&T`2022). - IEEE Ukraine Section. - Kharkiv National University of Radio Electronics
7. Risks Assessment and Approaches to Creative of the Reliable Software Modules for IoT Devices / Malinovskyi Vadym, Kupershtein Leonid, Lukichov Vitaliy . - Materials of International Conference on Innovative Solutions in Software Engineering. - November 29-30, 2022.- Ivano-Frankivsk, Ukraine.
8. Yuan Xiao, Yinqian Zhang, Radu Teodorescu. [Online]. Speech miner: a Framework for investigating and measuring speculative execution vulnerabilities [Electronic resource]: https://arxiv.org/pdf/1912.00329.pdf
9. Meltdown and Spectre: Which systems are affected by Meltdown?: [Electronic resource]: https://meltdownattack.com/#faq-systems-meltdown
10. Meltdown and Spectra: Which systems are affected by Meltdown?: [Electronic resource]: https://meltdownattack.com/#faq-systems-meltdown
11. Speculative Processor Vulnerability [Online]. ARM Developer Forum. Specifications Updated March 8, 2022 [Electronic resource]. – Режим доступу: https://developer.arm.com/Arm%20Security%20 Center/Speculative%20Processor%20Vulnerability
12. Cache Speculation Side-channels white paper [Online]. ARM Developer Forum. Specifications Updated March 8, 2022 [Electronic resource: https://developer.arm.com/documentation/102816/0205/
13. Kernel Side-Channel Attack using Speculative Store Bypass - CVE-2018-3639 [Electronic resource]: https://access.redhat.com/security/vulnerabilities/ssbd .
14. ISO/IEC, «Information technology — Security techniques-Information security risk management» ISO/IEC FIDIS 27005:2008.
15. Kakareka, Almantas (2009). 23. У Vacca, John. Computer and Information Security Handbook. Morgan Kaufmann Publications. Elsevier Inc. c. 393. ISBN 978-0-12-374354-1.
16. Serdar Yegulalp Rowhammer hardware bug threatens to smash notebook security / by Serdar Yegulalp// – March 9 – 2015 [Electronic resource]: https://www.infoworld.com/article/2894497/rowhammer-hardware-bug-threatens-to-smash-notebook-security.html
17. Kuljit Bains Patent US 20140059287 A1: Row hammer refresh command, February 27, 2014, by Kuljit Bains et al. [Electronic resource]: https://patents.google.com/patent/US20140059287
18. Cisco Systems security advisory. Row Hammer Privilege Escalation Vulnerability, // Cisco Systems security advisory. - March 11. – 2015 [Electronic resource]: Cisco Systems
19. S.G., Antonov, S.M. Klymov. Methodology for assessing the risks of disruption of the stability of the functioning of software and hardware complexes in the conditions of informational and technical effects // Nadezhnost. – 2017. – Volume 17. – No. 1. - P.32-39.
20. Sudhakar Govindavajhala and Andrew W. Appel. Using Memory Errors to Attack a Virtual Machine. // Princeton Edu University press – March 6.–2003. – [Electronic resource]: https://www.cs.princeton.edu/~appel/ papers/memerr.pdf.
21. M.V. Kartashov. Probability, processes, statistics. – Kyiv: VOC Kyiv University, 2007. – 504 p.
22. Threats and vulnerabilities of wireless networks. [Electronic resource] http://dspace.kntu.kr.ua/jspui/bitstream/123456789/5022/1/AUConferenceCyberSecurity_November2016_p 146.pdf
23. Analysis of protection mechanisms and vulnerabilities of wireless Wi-Fi networks. [Electronic resource]: http://ir.nmu.org.ua/butstream/handle
24. M. Swanson. NIST Special Publication 800-34 Rev. 1 Contingency Planning Guide for Federal Information Systems / M. Swanson, P. Bowen, A. W. Phillips, D. Gallup, D. Lynes. – 2010. – 149 p.
–: URL: https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2022/paper/view/15000.

Надійшла до редакції 1.12.2022 р.

МАЛІНОВСЬКИЙ ВАДИМ ІГОРЕВИЧ – к.т.н., доцент кафедри Захисту інформації, Вінницький національний технічний університет, м. Вінниця, Україна, ***e-mail: vad.malinovsky@gmail.com***

КУПЕРШТЕЙН ЛЕОНІД МИХАЙЛОВИЧ – к.т.н., доц., доцент кафедри захисту інформації, Вінницький національний технічний університет, м. Вінниця, Україна, ***e-mail: kupershtein@vntu.edu.ua***

КАПЛУН ВАЛЕНТИНА АПОЛІНАРІЄВНА – старший викладач кафедри захисту інформації, Вінницький національний технічний університет, м. Вінниця, Україна,
e-mail: valentina.kaplun@vntu.edu.ua