

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ МОРСЬКИЙ УНІВЕРСИТЕТ

Навчально-науковий інститут інформаційних технологій та
інноваційного підприємництва

Кафедра кібербезпеки та захисту інформації

Конспект лекцій

з дисципліни

КІБЕРФІЗИЧНІ СИСТЕМИ, ЇХ АРХІТЕКТУРА І ЗАХИСТ

для здобувачів

другого (магістерського) рівня вищої освіти
спеціальності F5 Кібербезпека та захист інформації
галузі знань F Інформаційні технології

Одеса-2026

Розробник: Кобозєва Алла Анатоліївна, доктор технічних наук, професор, завідувачка кафедри «Кібербезпека та захист інформації»

Конспект лекцій схвалено на засіданні кафедри «Кібербезпека та захист інформації»

(Протокол від «29» квітня 2026 р. №7)

Конспект лекцій схвалено на засіданні НМК ННІ ІТІП

(Протокол від «29» квітня 2026 р. № 7)

ЗМІСТ

ВСТУП	4
Тема 1. ПОНЯТТЯ КІБЕРФІЗИЧНОЇ СИСТЕМИ	6
Тема 2. ЦІЛЬОВІ АТАКИ НА КІБЕРФІЗИЧНІ СИСТЕМИ	25
Тема 3. СПЕЦИФІКА І ПРОБЛЕМИ ОРГАНІЗАЦІЇ КІБЕРЗАХИСТУ КІБЕРФІЗИЧНИХ СИСТЕМ РЕАЛЬНОГО ЧАСУ	38
Тема 4. КІБЕРФІЗИЧНІ СИСТЕМИ: АДАПТАЦІЯ ІСНУЮЧИХ СИСТЕМ ЗАХИСТУ ТА НОВІ КРИТЕРІЇ СТІЙКОСТІ НА ПРИКЛАДІ МОРСЬКИХ АВТОНОМНИХ ОБ'ЄКТІВ	50
Література	62

ВСТУП

Сьогодні еволюція інформаційних технологій досягла етапу, коли цифрові обчислювальні процеси нерозривно злилися з фізичним світом. Кіберфізичні системи (КФС) - такі як «розумні» енергомережі (Smart Grid), системи промислової автоматизації (ICS/SCADA), автономні транспортні засоби, робототехніка та інфраструктура «розумного міста» - формують хребет сучасної критичної інфраструктури. На відміну від класичних ІТ-систем, де головними пріоритетами є конфіденційність та цілісність даних, у КФС на перше місце виходить безпека фізичних процесів, безперервність функціонування та реагування в реальному часі. Помилка в коді або збій у мережі датчиків тут загрожують не просто втратою файлів, а реалістичними техногенними катастрофами, людськими жертвами та колапсом цілих галузей економіки.

Традиційні підходи до кібербезпеки тут можуть виявитися недостатньо ефективними або неможливими до застосування. Обмежені обчислювальні ресурси мікроконтролерів і промислових контролерів, жорсткі вимоги до затримок у реальному часі тощо створюють унікальний ландшафт загроз. Злочинці та деструктивні актори все частіше фокусуються на атаках на фізичний стан систем (через маніпуляції датчиками, компрометацію виконавчих механізмів чи впровадження шкідливого мікропрограмного забезпечення), що робить дисципліну «Кіберфізичні системи, їх архітектура і захист» актуальною та затребуваною.

Метою дисципліни є розвиток у здобувачів здатності: розуміти архітектуру, принципи побудови та функціонування сучасних КФС; аналізувати вразливості та вектори кібератак на критичну інфраструктуру й автономні системи (на прикладі сценаріїв MASS); застосовувати методи захисту й стеганографії в контурах управління КФС з урахуванням жорстких часових обмежень реального часу.

Завданнями, що мають бути вирішеними у процесі викладання освітнього компонента (навчальної дисципліни), є набуття здобувачами компетентностей за освітньою програмою.

Загальні компетентності:

КЗ 1. Здатність застосовувати знання у практичних ситуаціях.

Фахові (спеціальні) компетентності:

КФ1. Здатність обґрунтовано застосовувати, інтегрувати, розробляти та удосконалювати сучасні інформаційні технології, фізичні та математичні моделі, а також технології створення та використання прикладного і спеціалізованого програмного забезпечення для вирішення професійних задач у сфері інформаційної безпеки та/або кібербезпеки.

КФ3. Здатність досліджувати, розробляти і супроводжувати методи та засоби інформаційної безпеки та/або кібербезпеки на об'єктах інформаційної діяльності та критичної інфраструктури.

КФ4. Здатність аналізувати, розробляти і супроводжувати систему управління інформаційною безпекою та/або кібербезпекою організації,

формувати стратегію і політики інформаційної безпеки з урахуванням вітчизняних і міжнародних стандартів та вимог.

КФ8. Здатність досліджувати, розробляти, впроваджувати та супроводжувати методи і засоби криптографічного та технічного захисту інформації на об'єктах інформаційної діяльності та критичної інфраструктури, в інформаційних системах, а також здатність оцінювати ефективність їх використання, згідно встановленої стратегії і політики інформаційної безпеки та/або кібербезпеки організації.

Вивчення освітнього компонента (навчальної дисципліни) передбачає здобуття таких *результатів навчання*:

РН3. Проводити дослідницьку та/або інноваційну діяльність в сфері інформаційної безпеки та/або кібербезпеки, а також в сфері технічного та криптографічного захисту інформації у кіберпросторі.

РН6. Аналізувати та оцінювати захищеність систем, комплексів та засобів кіберзахисту, технології створення та використання спеціалізованого програмного забезпечення.

РН13. Досліджувати, розробляти, впроваджувати та використовувати методи та засоби криптографічного та технічного захисту інформації бізнес/операційних процесів, а також аналізувати і надавати оцінку ефективності їх використання в інформаційних системах, на об'єктах інформаційної діяльності та критичної інфраструктури.

РН14. Аналізувати, розробляти і супроводжувати систему аудиту та моніторингу ефективності функціонування інформаційних систем і технологій, бізнес\операційних процесів у сфері інформаційної та/або кібербезпеки в цілому.

РН21. Використовувати методи натурного, фізичного і комп'ютерного моделювання для дослідження процесів, які стосуються інформаційної безпеки та/або кібербезпеки.

Тема 1. ПОНЯТТЯ КІБЕРФІЗИЧНОЇ СИСТЕМИ

План

1. Визначення кіберфізичної системи. Категорії кіберфізичних систем за типом ресурсів
2. Еволюція інженерних рішень
3. Фундаментальні властивості кіберфізичної системи
4. Сфери використання кіберфізичних систем
5. Відмінності між ІТ та КФС
6. Аналогія КФС та людини
7. Взаємодія обчислювального ядра з апаратним середовищем: сенсори, інтерфейси та промислові контролери

1. Визначення кіберфізичної системи. Категорії кіберфізичних систем за типом ресурсів

На міжнародному рівні (зокрема у стандартах NIST (Національний інститут стандартів і технологій США)) кіберфізичні системи визначаються так:

Кіберфізичні системи - це інтегровані цифрові, аналогові, фізичні компоненти, об'єднані для спільного функціонування за допомогою інтегрованої фізики та логіки через *безперервний цикл зворотного зв'язку*.

Якщо перекласти це на людську мову: Кіберфізична система (КФС) - це не просто комп'ютер чи телефон. Це технічний комплекс, де цифровий «мозок» (програми, алгоритми, мікроконтролери) і фізичне «тіло» (датчики, механізми, двигуни) злилися в єдине ціле і працюють рука об руку в реальному часі.

Простими словами: це трикутник, де **обчислення**, мережі **зв'язку** та **фізичні процеси** утворюють єдиний організм. Якщо вилючити хоча б один елемент (наприклад, забрати зв'язок чи обчислення), система перетворюється на звичайний шматок заліза.

Умовно всі КФС можна розділити на дві великі категорії за типом ресурсів:

- **Мобільні та периферійні КФС:** безпілотники (БПЛА), автономні судна (MASS), робототехніка, транспортні засоби та розподілені датчики (IoT);
- **Стаціонарні та великомасштабні промислові КФС:** розумні енергомережі (Smart Grid), системи керування атомними станціями, великі хімічні заводи чи автоматизовані лінії на виробництвах.

Приклади:

- Сучасний автомобіль: це не просто залізо з колесами, а десятки мікрокомп'ютерів, які постійно контролюють тиск у шинах, швидкість обертання коліс, гальма та впорскування палива;
- Світлофор на перехресті: він реагує на потік машин під асфальтом і сам вирішує, коли вмикати зелене світло.

- Промисловий робот на заводі: безперервно зчитує положення деталей і ювелірно точно зварює деталі кузова.
- Сучасне судно або порт: автоматизовані крани в порту самі розраховують траєкторію переміщення контейнерів, а на суднах системи динамічного позиціонування (Dynamic Positioning) у реальному часі керують підрулюючими пристроями та гвинтами, щоб утримувати корабель на точних координатах під час шторму без кидання якоря.
- Автопілот автомобіля чи судна: система не просто «знає маршрут», вона за мілісекунди реагує на зміну обставин навколо, бо фізичний світ не прощає запізнень.

2. Еволюція інженерних рішень

Щоб зрозуміти унікальність КФС, корисно подивитися, як інженери вирішували подібні завдання раніше.

1. Епоха великих ЕОМ (до 1980-х): обчислення поза контуром реального часу

У ту епоху великі комп'ютери вже залучалися до розв'язання складних завдань для реального світу, наприклад, для балістичних розрахунків, проєктування суден чи аналізу напружень у конструкціях літаків.

Проте тодішні комплекси ще не були кіберфізичними системами через технічні обмеження архітектури:

- *Відсутність єдиного контуру реального часу*: обробка даних була розірваною (пакетна обробка). Інженери збирали параметри у фізичному світі, переносили їх на перфокарти, завантажували в ЕОМ, годинами чекали на результат роздруківки, а потім застосовували ці дані на виробництві. Комп'ютер не був вбудований у безперервний процес обміну сигналами;
- *Фізичні бар'єри (габарити та інтерфейси)*: ЕОМ того часу споживали величезну кількість енергії, виділяли купу тепла і займали цілі зали. У них просто не було можливості тісно інтегруватися з датчиками на рухомих об'єктах чи в умовах виробничого цеху.

Приклад. Коли проєктували корабель у 1960–70-х, ЕОМ обчислювала гідродинаміку корпусу за заздалегідь зібраними таблицями. Але під час ходових випробувань саме судно керувалося механічними тягами чи простими аналоговими приладами. Комп'ютер допомагав розраховувати, але не керував об'єктом у прямому ефірі.

2. Епоха ПК та інтернету (1990–2010).

Комп'ютер є в кожного на столі, потім у кишені (смартфон). Ми передаємо інформацію, листуємося, купуємо в інтернет-магазинах. Але комп'ютер усе ще переважно працює з *інформацією*, а не рухає важкі предмети в реальному світі.

У цей період технології зробили величезний стрибок: комп'ютери переїхали з величезних залів на робочі столи інженерів, а згодом з'явився масовий інтернет і локальні мережі. Зміни на практиці:

- *Проектування відбувається за допомогою ПК (CAD-системи замість кульманів)*: інженери-суднобудівники більше не креслять корпуси суден

ручками на ватмані. На столах з'являються персональні комп'ютери з програмами тривимірного проєктування (у суднобудуванні це, наприклад, спеціалізовані пакети на кшталт *Tribon* чи *AutoCAD*); корабель спочатку створюється як цифрова 3D-модель у комп'ютері, а вже потім моделі передаються на заводи.

- *Виникає електронна документація:* креслення та розрахунки більше не возять пачками потягами чи не відправляють поштою тижнями. Проєктні файли перекидають через перші локальні мережі підприємств, корпоративну пошту або записують на компакт-диски та дискети й відправляють адресату.
- *Що відбувається на самому судні?* У машинному відділенні та на містку збудованого корабля вже активно працюють комп'ютери та контролери. Вони керують дизель-генераторами, слідкують за рівнем палива в цистернах або виводять цифрові карти на екрани навігаторів. Але ці комп'ютерні системи на борту судна повністю ізольовані. У них є своя локальна мережа кабелів усередині корабля, але немає жодного виходу в глобальний інтернет чи зв'язку з береговим офісом у режимі реального часу.

Таким чином, у період 1990–2010 років комп'ютери вже керували багатьма процесами на судні, але судно все ще залишалося «острівцем безпеки» від кібератак. Навіть якщо зловмисник хотів нашкодити кораблю, який перебуває в океані, він фізично не міг до нього «дотягнутися», бо не було супутникового інтернету високої швидкості на кожному судні, не було хмарних сервісів і віддаленого доступу техпідтримки з берега. Усі збої та проблеми вирішувалися силами механіків на борту, які перезавантажували завислий контролер або міняли згорілу плату.

3. *Епоха КФС (сучасність)*

Згодом мікропроцесори стали крихітними, дешевими й енергоефективними, і лише тоді з'явилася можливість «впаяти» обчислювальний центр безпосередньо у фізичний об'єкт, переходячи до КФС, ставлячи питання: добрим чи поганим є цей перехід, оскільки роблячи фізичний об'єкт «розумним» і цифровим, ми автоматично відкриваємо ящик Пандори і робимо його вразливим для хакерів, збоїв софту та кібератак, чого раніше не було.

До появи мікропроцесорів паровий котел, судновий рушій чи автомобільні гальма були «нерозумними», але механічно надійними й абсолютно недоступними для хакера з іншого кінця планети.

То чому ж людство добровільно пішло на цей ризик? Чого не вистачало чистому «механічному» світу і чому знадобилося вбудовувати туди комп'ютери?

Розглянемо **фундаментальні причини**, які змусили інженерів об'єднати залізо з цифрою:

1. Старі аналогові системи працювали за жорсткими фізичними правилами (шестерні, важелі, електричні реле). Якщо умови зовнішнього

середовища змінювалися так, як інженери не передбачили під час проєктування, система ламалася або зупинялася.

Приклад 1. Звичайний автомобільний карбюратор механічно подає суміш. Але якщо ви підіймаєтесь у гори на висоту 3000 метрів, повітря розріджується, і двигун починає «задохатися», втрачаючи потужність.

Мікроконтроллер усередині сучасного двигуна зчитує тиск і кисень через датчики і щомиті змінює налаштування «на льоту».

Таким чином, механіка є жорсткою і «не вміє думати», створюючи брак адаптивності.

2. *Незадовільна швидкість реакції людини та аналогових схем:* фізичний світ змінюється блискавично, а людська реакція у критичній ситуації становить 1–1.5 секунди. На швидкості 100 км/год за цей час машина проїжджає майже 40 метрів.

Антиблокувальна система гальм або система стабілізації у сучасному авто приймають рішення за мілісекунди, імпульсно натискаючи на гальма окремих коліс до 15 разів на секунду. *Жодна людина і жодна класична механіка на таке не здатні.*

3. Сучасний світ вимагає шаленої економії ресурсів, максимум ефективності. Електростанції, заводи, літаки мають працювати на межі своїх можливостей, але без аварій. Комп'ютер може контролювати мільйони параметрів одночасно (температуру в кожній точці турбіни, напругу, вібрацію) і балансувати систему так тонко, як це не зробить жоден оператор за пультам. Це економить мільярди тонн палива і мільйони годин роботи.

4. *Автономія та порятунок людських життів:* є речі, де присутність людини неможлива або смертельно небезпечна: глибоководні апарати, марсоходи, дрони-розвідники, ліквідація аварій на атомних станціях. Без вбудованого цифрового «мозку» такі апарати просто не могли б існувати.

АЛЕ: ефективність, швидкість і автономність завжди воюють із безпекою.

Чим більше ми комп'ютеризуємо фізичний світ, тим більшу «поверхню атаки» створюємо. Раніше, щоб зупинити завод, диверсанту треба було фізично зайти в цех і перекрити вентиль. Сьогодні достатньо запуснути шкідливий код через офісний комп'ютер бухгалтерії, який якимось чином мав мікроскопічний «місток» до виробничої мережі.

Повернемося до *прикладу* з судном. Сучасне судно підключають до супутникового інтернету (Starlink тощо), берегові офіси моніторять роботу двигунів у реальному часі з іншого континенту, а навігаційні системи оновлюють карти «по повітрю». І ось тут відкривається поверхня атаки: тепер хакер може теоретично зламати супутникові канали зв'язку або береговий сервер компанії та втрутитися в роботу судна дистанційно.

Саме тому наш курс і має назву «...і захист». Ми не можемо відмовитися від КФС, бо без них зупиниться сучасна цивілізація (не полетять літаки, зупиняться порти, зникне світло). Але ми мусимо навчитися будувати їх так, щоб мінімізувати ці вразливості.

3. Фундаментальні властивості кіберфізичної системи

Чим кіберфізична система кардинально відрізняється від звичайного софту (який крутиться в хмарі чи на телефоні) або від звичайної глухої механіки (як старий годинник чи поршневий насос)?

Вона має *три обов'язкові фундаментальні властивості*:

1. Замкнений цикл зворотного зв'язку

Звичайна програма на комп'ютері найчастіше працює лінійно: ви натиснули кнопку, вона виконала дію і зупинилася в очікуванні наступної команди людини. КФС так не працює. Вона «живе» в безперервному нескінченному колі дій і спостережень. На практиці це виглядає так (алгоритм циклу):

Датчик фіксує стан світу → Цифровий мозок обробляє дані й ухвалює рішення → Виконавчий механізм робить фізичну дію → Фізичний світ змінюється → Датчик знову фіксує новий стан (і все повторюється знову, тисячі разів на секунду).

Такий алгоритм роботи КФС є надзвичайно важливим: система постійно перевіряє, чи принесла її дія бажаний результат. Якщо зовнішні умови змінилися (наприклад, подув бічний вітер або змінився тиск), цикл миттєво підлаштовує команду.

Приклад (система динамічного позиціонування судна): Судно знаходиться в океані на точці під час бурі без якоря. Мережа датчиків GPS і гіроскопів постійно «бачить», що хвилі зносять корабель на 2 метри ліворуч. Комп'ютер блискавично обчислює зсув і дає команду підрулюючим гвинтам повернути судно назад. Щойно судно вирівнюється, датчики це підтверджують, і цикл робить наступну ітерацію. Без цього неперервного циклу судно знесе.

2. Робота в реальному часі

У світі КФС поняття «часу» має зовсім іншу вагу, ніж у звичайному ІТ:

- *У звичайному ІТ:* якщо завис браузер або сторінка в інтернет-магазині відкривається на 2 секунди довше через затримку мережі, це викликає дратування. Трагедії не сталося.
- *У кіберфізичних системах:* затримка на 2 мілісекунди в команді екстреного гальмування автомобіля, розгортання подушки безпеки чи закриття аварійного клапана на хімічному заводі означає катастрофу. У КФС обчислення мають бути не просто «правильними», вони мають бути виконані вчасно. Якщо результат пораховано занадто пізно, він стає абсолютно непотрібним (як парашут, який відкрився вже після удару об землю).

Приклад (антиблокувальна система коліс — ABS): коли ви різко гальмуєте на льоду, колеса можуть заблокуватися, і машина стане неуправляною. Комп'ютер КФС повинен «зловити» момент зриву зчеплення, послабити тиск у гальмах і відновити його до 15 разів на секунду. Затримка хоча б у чверть секунди означає, що автомобіль уже летить у кювет.

3. Тісний і нерозривний зв'язок «цифри» і «фізики»

У КФС цифровий код і фізичне залізо поділяють спільну долю. Вони утворюють симбіоз, де пошкодження однієї складової миттєво руйнує іншу. Це проявляється у двох напрямках:

- **Помилка в цифровому коді ламає фізичний світ:** якщо програміст схибив у логіці керування роботом-маніпулятором на конвеєрі (наприклад, переплутав координати або допустив зависання потоку), віртуальний код не просто «викине синій екран смерті». Робот на повній швидкості вдарить металевою рукою в стіну, розтрощить обладнання або травмує людину.
- **Фізична поломка руйнує цифрову логіку:** якщо у фізичному світі сталося щось прозаїчне, наприклад, порвався кабель передачі даних від датчика температури, туди потрапила волога або пацюк перегриз дріт, то цифровий мозок залишається «сліпим». Бачачи нульові або хибні показники, програма може вирішити, що котел охолоджується, і подати туди максимальну кількість палива, спричинивши вибух.

Висновок. У КФС неможливо вивчати окремо «програмування» і окремо «механіку». Будь-який кібер-інцидент тут має фізичний наслідок, і навпаки.

4. Сфери використання КФС

КФС вже давно перестали бути фантастикою, сьогодні вони утворюють непомітний, але критичний каркас нашої цивілізації. Якщо вимкнути ці системи, сучасні міста й економіки зупиняться за лічені хвилини. Ось головні сфери їх застосування:

1. Транспорт та логістика.

Транспортна галузь сьогодні - це суцільні кіберфізичні системи, де людський фактор зводиться до мінімуму задля безпеки та швидкості.

- **Автомобільний транспорт:** сучасні автомобілі (особливо електромобілі та автопілоти) мають десятки комп'ютерів, які керують усім: від екстреного гальмування та стабілізації на льоду до аналізу перешкод на дорозі через камери й лідари;
- **Морський транспорт та порти:** сучасні контейнеровози обладнані системами динамічного позиціонування. Коли судно підходить до платформи чи вузького порту, комп'ютер сам керує гвинтами й підрулюючими пристроями, утримуючи судно з точністю до сантиметра попри шторм і вітер. У самих портах крани-автомати самі розвантажують величезні судна за заздалегідь прорахованими алгоритмами;
- **Залізниця:** швидкісні потяги керуються автоматикою, яка не дозволить машиністу проїхати на червоне світло чи перевищити швидкість на повороті: система сама загальмує потяг, якщо людина проігнорує попередження.

2. Енергетика та критична інфраструктура.

Енергетичні мережі - це найбільші і найскладніші КФС на планеті.

- *«Розумні електромережі» (Smart Grid)*: якщо через шторм десь обірвалося дерево і сталася аварія на лінії, стара мережа просто гасила цілий регіон. Сучасна Smart Grid за мілісекунди автоматично перерозподіляє потоки струму альтернативними шляхами, ізолюючи пошкоджену ділянку, доки аварійні бригади ще їдуть на місце;
- *Вітрові та сонячні електростанції*: комп'ютери постійно крутять важкі сонячні панелі слідом за сонцем або змінюють кут лопатей гігантських вітряків залежно від сили й напрямку вітру, вичавлюючи максимум енергії і захищаючи конструкцію від руйнування при бурі.

3. Медицина

Це найделікатніша і водночас найвразливіша область КФС, адже тут «фізичним тілом» є жива людина.

- *Імплантовані кардіостимулятори*: це крихітні комп'ютери з датчиками всередині серця. Вони постійно вимірюють серцевий ритм, і якщо фіксують небезпечну зупинку чи аритмію, подають мікроскопічний електричний імпульс, змушуючи серце скорочуватися правильно;
- *Інсулінові помпи*: пристрої, які працюють у парі з датчиком рівня цукру в крові. Вони автоматично вимірюють рівень глюкози і в потрібний момент вводять необхідну дозу інсуліну без участі пацієнта.

4. Промисловість: індустрія 4.0 (розумні заводи)

Заводи більше не нагадують темні й шумні цехи минулого століття з робітниками біля кожного верстата.

- *Автоматизовані конвеєри та робот-маніпулятори*: роботи на автомобільних чи електронних заводах зварюють деталі, фарбують кузови чи збирають мікросхеми з ювелірною точністю до міліметра;
- *Предиктивна аналітика*: датчики вібрації та температури на промислових насосах чи турбінах аналізують стан заліза в реальному часі. Якщо підшипник починає зношуватися і вібрує інакше, ніж зазвичай, система сама каже інженеру: *«Цей вузол зламається через три дні, заплануйте заміну під час нічної перерви»*, запобігаючи аварії.

5. «Розумний» будинок та місто (Smart City)

- *Міська інфраструктура*: «розумні» світлофори реагують на кількість машин на під'їзді до перехрестя, оптимізуючи рух і зменшуючи затори. Системи контролю води автоматично перекривають магістралі при прориві труб;
- *Побут*: звичайні датчики протікання води під раковиною, які за секунду за допомогою електрокрана перекривають загальний стояк, рятуючи квартиру від затоплення ще до того, як господар це помітить.

Таким чином, КФС - це не якась абстрактна галузь майбутнього, а те, що оточує нас щодня у найрізноманітніших проявах.

5. Відмінності між ІТ та КФС

Студенти часто плутають звичайне ІТ (інформаційні технології) та ОТ (операційні технології, до яких належать КФС).

Операційні технології (ОТ) - це апаратно-програмні комплекси, які безпосередньо керують, моніторять або змінюють фізичні пристрої, процеси та події у реальному світі.

Якщо ІТ працюють із віртуальними даними (файли, бази даних, вебсайти, фінансові транзакції), то ОТ працюють із матеріальним залізом (клапани, двигуни, конвеєри, турбіни, насоси).

Класичний приклад ОТ до епохи інтернету. Промисловий контролер на заводі, який за допомогою реле відкриває клапан подачі води. У нього немає операційної системи Windows, немає браузера, немає антивірусу. Він виконує одну функцію: «якщо тиск перевищив норму, то перекрий кран». Це є класична операційна технологія.

Кіберфізичні системи є новим, вищим етапом розвитку ОТ. Щоб зрозуміти цей зв'язок, розглянемо еволюцію:

- *Традиційне ОТ (минуле століття):* закриті, ізольовані промислові системи керування (промислові контролери, датчики, локальні шини). Вони керували фізичним світом, але були «нерозумними» і відірваними від глобальних мереж;
- *Кіберфізичні системи (сучасність):* це та ж сама ОТ, але пов'язана з «мозком» нового покоління (складними алгоритмами, штучним інтелектом, хмарними обчисленнями) та підключена до мереж зв'язку.

Тобто, КФС - це «розумне», мережеве ОТ. Вони успадкували від операційних технологій головне - відповідальність за управління фізичним процесом, але додали до нього цифрову гнучкість і комунікацію.

Коли ми говоримо про кібербезпеку КФС, ми насправді говоримо про безпеку ОТ - захист фізичної інфраструктури від цифрових загроз.

Розглянемо порівняння між ІТ та КФС:

Критерій	Звичайне ІТ (сайти, бази даних, офіси)	КФС / ОТ (промисловість, авто, транспорт)
Головна мета	Зберегти, обробити та захистити інформацію.	Керувати фізичним процесом у реальному часі.
Цінність для хакера	Вкрасти паролі, номери карт, персональні дані.	Зупинити завод, пустити поїзд під укіс, вимкнути світло в місті.
Ціна помилки	Фінансовий збиток, репутація.	Зірване життя, аварії, екологічні катастрофи.
Швидкість реакції	Секунди або хвилини (перезавантажив сервер).	Мілісекунди (миттєва реакція на зміну фізики).
Життєвий цикл	3–5 років (оновив софт чи пристрій).	10–30 років (турбіна чи судновий двигун працюють десятиліттями).

Розуміння того, що КФС - це насамперед операційні технології, а не звичайне ІТ, повністю змінює філософію захисту та проектування.

Головна небезпека полягає в тому, що класичні ІТ-фахівці часто намагаються керувати кіберфізичними системами за допомогою звичних ІТ-методів. Наприклад, запустити планове оновлення операційної системи на сервері підприємства або перезавантажити контролер посеред робочого циклу. Для ІТ це звична рутина. Але в світі ОТ подібне «невинне» перезавантаження під час польоту літака, роботи електростанції чи керування судновим двигуном може призвести до миттєвої втрати керованості та аварії.

Саме тому в кіберфізичних системах діє інша ієрархія пріоритетів безпеки (так звана тріада СІА розвертається в протилежному напрямі):

1. *Доступність*: система повинна працювати 24/7 без зупинок;
2. *Цілісність*: дані датчиків і команди мають бути достовірними;
3. *Конфіденційність*: захист від витоку даних (який у класичному ІТ завжди на першому місці, у світі ОТ часто відходить на задній план).

У світі ОТ діє залізне правило: безпека фізичного процесу та безперервність роботи завжди важливіші за віртуальну конфіденційність.

6. Аналогія КФС та людини

Щоб назавжди зрозуміти й легко уявляти архітектуру будь-якої кіберфізичної системи (чи то автономне судно, чи то робот на заводі, чи то звичайний кондиціонер тощо), достатньо подивитися на власне тіло. *Людина - це найдосконаліша у світі природна кіберфізична система.*

Розберемо цю аналогію детально:

1. Датчики (очі, вуха, шкіра та рецептори) - рівень сприйняття

У тілі людини рецептори постійно вимірюють зовнішній і внутрішній світ: очі бачать світло, вуха ловлять звуки, рецептори на шкірі відчують температуру, а внутрішні органи сигналізують про тиск чи біль.

У світі КФС - це сенсори (датчики). Вони перетворюють фізичні параметри на «мову комп'ютера» - електричні сигнали, цифри, нулі та одиниці.

Приклади:

- *Термометр чи термopара* (шкіра, що відчуває холод чи спеку).
- *Камери, лідари та радарні датчики* на автомобілі чи судні (очі, що бачать перешкоди).
- *Датчики тиску та вібрації* на промисловій турбіні (внутрішні відчуття напруження).

Якщо датчик зламався або його «обдурили» (наприклад, засліпили камеру лазером), система втрачає адекватне уявлення про реальність, подібно до того, як людина з зав'язаними очима не розуміє, куди йде.

2. Контролер / процесор (головний мозок) - рівень обробки

Коли ви випадково торкаєтеся гарячої праски, сигнал від рецепторів шкіри блискавично летить у спинний і головний мозок. Мозок аналізує інформацію: *«Це небезпечно, треба негайно реагувати!»*

У світі КФС: це мікроконтролери, промислові контролери або вбудовані комп'ютери. Вони діють як цифровий мозок: отримують потоки цифр від датчиків, порівнюють їх із закладеними правилами чи алгоритмами і приймають рішення.

Мозок КФС не має часу на довгі роздуми чи «зависання». Він працює за жорсткими правилами реального часу, інколи ухвалюючи тисячі рішень на секунду.

3. Виконавчі механізми / актуатори (м'язи, руки та ноги) - рівень дії

Отримавши команду від мозку («відсмикнути руку!»), нервова система передає імпульс у м'язи. М'язи скорочуються, і рука виконує фізичну дію у реальному світі.

У світі КФС: це актуатори (виконавчі механізми). Вони перетворюють цифрову команду мікроконтролера на реальну фізичну роботу: рух, силу, тепло чи тиск.

Приклади:

- *Електродвигуни та сервоприводи* (які крутять колеса або повертають стерно судна).
- *Клапани та соленоїди* (які відкривають чи перекривають подачу газу, води або палива під високим тиском).
- *Нагрівальні елементи чи охолоджувачі.*

Щоб остаточно зрозуміти аналогію, простежимо за одним циклом життя кіберфізичної системи на дорозі:

1. *Датчик (очі):* радар фіксує, що машина попереду різко загальмувала.
2. *Мозок (нервова система):* блок управління двигуном і гальмами миттєво рахує відстань і швидкість зближення. *Висновок: аварія неминуча, якщо не загальмувати.*
3. *Актуатор (м'язи):* мозок посиляє електричний сигнал на гідравлічний клапан гальмівної системи, який з величезною силою тисне на гальмівні колодки.
4. *Зворотний зв'язок:* датчики швидкості обертання коліс одразу повідомляють мозку, що машина почала сповільнюватися. Цикл повторюється знову, доки авто повністю не зупиниться.

Ця біологічна аналогія допомагає одразу зрозуміти найголовніше: технічна система така ж вразлива, як і живий організм. Пошкодьте «очі» (датчики), переріжте «нерви» (кабелі зв'язку) або затьмарьте «мозок» (зламайте програму), і потужна фізична машина стане некерованою.

ВИСНОВОК: ми живемо в епоху, коли цифра і фізика остаточно злилися. Якщо раніше комп'ютери були просто інструментами для підрахунків, які стояли в закритих кабінетах, то сьогодні крихітні мікроконтролери та глобальні мережі вбудовані у кожне коло, кожен двигун, кожен турбінний датчик. Завдяки цьому наші автомобілі стали безпечнішими, заводи - ефективнішими, а медичне обладнання - здатним рятувати життя на відстані.

Але за цей технологічний прорив доводиться платити високу ціну. Роблячи фізичний світ «розумним», ми водночас робимо його вразливим. У

традиційному ІТ найстрашніше, що може статися під час зльоту системи, це втрата небереженого документа чи витік паролів. У світі КФС (операційних технологій) будь-який збій у програмі чи зумисна кібератака перетворюється на реальну фізичну катастрофу: зупинку судна посеред шторму, вимкнення світла у цілому місті чи аварію на виробництві. Саме тому сучасний фахівець із кібербезпеки, захисту інформації не може просто сидіти за монітором і перевіряти антивіруси. Він має розуміти, як працює фізичний світ, чому залізо відрізняється від софту, і чому в кіберфізичних системах головним пріоритетом стає не просто збереження даних, а безпека життя та неперервність фізичних процесів.

7.Взаємодія обчислювального ядра з апаратним середовищем: сенсори, інтерфейси та промислові контролери

Уявіть собі складну проблему: цифровий процесор живе у світі абстрактних нулів і одиниць (наприклад, 0 вольт - це «0», а 3.3 або 5 вольт - це «1»). Фізичний же світ навколо нас - аналоговий, неперервний і різноманітний: температура плавно зростає, тиск у трубі коливається, а швидкість вітру змінюється кожної мілісекунди.

Як мікрочип у чорній коробочці може «побачити» пожежу, «відчути» вібрацію турбіни чи «повернути» колесо автомобіля?

Розберемо цей шлях крок за кроком.

Крок 1. Перший ланцюжок у будь-якій КФС - це перетворення фізичного світу на цифрову мову. За це відповідають датчики (сенсори) та їхні «помічники». Сам по собі датчик (наприклад, термопара чи п'єзоелемент) - це пристрій, який реагує на зміну фізичного середовища (температури, тиску, світла) і змінює свої електричні характеристики (опір, напругу). Але ця електрична напруга змінюється плавно і неперервно, а цифрові виводи процесора оперують лише дискретними станами (0 і 1). Сам датчик не передає процесору готове число «1.427 вольт», процесор його просто не зрозуміє, оскільки він розуміє лише «0» і «1». Тут у гру вступає мікросхема-перекладач - АЦП (аналогово-цифровий перетворювач). Вона бере електричний сигнал від датчика, вимірює його мільйони разів на секунду і перетворює на дискретний числовий код (наприклад, число 10110101).

Аналогія:

- Отримання цифрового зображення;
- Перетворення плавного музичного звуку на набір цифр у цифровому аудіозапису (процеси дискретизації та квантування).

Тепер цифровий «мозок» може «прочитати» цю цифру і зрозуміти: *«Ага, температура двигуна досягла 90 градусів».*

АЛЕ Коли ми перетворюємо неперервний аналоговий світ на цифру, ми завжди втрачаємо частину інформації. Ми намагаємося нескінченно детальний і плавний фізичний світ «заштовхнути» у скінченну сітку дискретних чисел. Це призводить до деяких негативних наслідків і вразливостей:

А) Технічні наслідки оцифровки:

- Похибка квантування (шум округлення): оскільки шкала АЦП скінченна (наприклад, 1024 рівні для 10-бітного АЦП), будь-яке значення між двома сходами округлюється. Мікроконтролер «не бачить» мікрозмін сигналу, які менші за один крок квантування. Для системи цих міні-коливань просто не існує.

- Пропуск швидких подій (Аліасінг): якщо АЦП оцифровує сигнал порівняно рідко (порушуючи теорему Котельникова/Найквіста), швидкі фізичні процеси між точками вимірювання повністю випадають. Система може пропустити короткий, але критичний стрибок тиску чи різкий удар вібрації. На жаль, на практиці цим правилом «нехтують» (або намагаються зекономити на ньому) набагато частіше, ніж хотілося б архітекторам і фахівцям з безпеки. Причини цього бувають як суто технічними, так і економічними:

1. Недооцінка фізики

Розробники (особливо ті, хто прийшов із класичного ІТ) часто міркують так: «Температура в циліндрі двигуна або рівень води в цистерні змінюються плавно, на кілька градусів чи сантиметрів за хвилину. Навіщо нам оцифровувати сигнал тисячі разів на секунду? Будемо знімати показник раз на секунду, цього ж головою вистачить!». Але вони забувають, що реальний світ не буває «ідеальним». Окрім повільної зміни температури, на дрід і датчик постійно діють високочастотні перешкоди: електромагнітні поля від силових кабелів, вібрації від роботи двигуна чи радіочастотні наведення. Якщо не фільтрувати і не оцифровувати частіше, ці високочастотні шуми через аліасінг «перевернуться» в низькочастотну область і створять у цифрі неіснуючі, але дуже правдоподібні коливання, якими замаскують реальну аварію.

2. Економія ресурсів

- Висока частота дискретизації вимагає потужнішого процесора (або контролера з високою тактовою частотою), більшого обсягу пам'яті для збереження масивів даних і більшої пропускну здатності шини.
- У масовому виробництві (наприклад, IoT-пристрої, датчики для авто) інженери намагаються зекономити скрізь. Вони беруть дешевший мікроконтролер із повільним АЦП: «Та і так зійде, головне, щоб базові цифри показував».

3. Дилема: швидкість керування проти фільтрації

Це найтонша інженерна проблема. Щоб захистити систему від аліасінгу, перед АЦП ставлять аналоговий фільтр (або застосовують цифрову фільтрацію). Але за законами фізики будь-який фільтр створює затримку сигналу в часі. У швидких кіберфізичних системах (наприклад, системах стабілізації літака, ракеті, чи динамічному позиціонуванні судна під час шторму) кожна мілісекунда затримки в управлінні є критичною. Якщо фільтр «думає» надто довго,

затримуючи сигнал на кілька мілісекунд, система керування почне запізнюватися і замість стабілізації може почати розгойдувати об'єкт (втратить стійкість). Іноді інженери свідомо жертвують якістю фільтрації та вибором надлишкової частоти дискретизації заради того, щоб прискорити реакцію контуру керування.

З погляду кібербезпеки аліасінг виступає як зброя зловмисника, і коли розробники закривають очі на ці нюанси заради економії чи швидкості, вони створюють ідеальну вразливість для цілеспрямованих атак. Якщо зловмисник знає, що в системі немає належного захисту від високочастотних перешкод (або частота оцифровки занадто низька), він може дистанційно або через контакт із проводкою накласти на датчик спеціально підібраний високочастотний сигнал. Цей сигнал неможливо зафіксувати напряду (бо він поза робочим діапазоном), але завдяки ефекту накладання спектрів (аліасингу) він перетвориться в контролері на плавний, «безпечний» зсув показників. Система повірить, що все гаразд, хоча фізичний вузол уже руйнується.

Б) Чому це критично для кібербезпеки КФС?

Для фахівця з кібербезпеки ці обмеження оцифровки перетворюються на вразливі типи атак, якими активно користуються зловмисники:

- *«Сліпа зона» системи та атаки на малі зміни:* оскільки комп'ютер оперує лише дискретними значеннями, він має так звану «мертву зону чутливості». Зловмисник може непомітно для системи вносити малі зміни у фізичне середовище (наприклад, повільно й непомітно підвищувати температуру чи змінювати тиск у межах одного кроку квантування або трохи вище). Цифровий мозок деякий час не помічатиме цього, бо для нього показники залишаються «в межах норми», тоді як фізичний об'єкт наближається до руйнування.

- *Маніпуляція через фізичні спотворення:* кіберфізичні системи повністю залежать від правила: «Що прийшло в цифрі на порт, те і правда». Контролер за визначенням вірить отриманому числу. Якщо зловмисник знає характеристики АЦП і частоту дискретизації, він може вплинути на аналоговий дріт зовні (наприклад, накласти електромагнітні перешкоди певної частоти або акустичний резонанс на датчик детонації двигуна). АЦП інтерпретує цей шум так, що на виході утвориться цілком конкретне «фальшиве» число. Процесор повірить у це число і почне приймати хибні рішення (наприклад, аварійно зупиняти справний двигун або, навпаки, ігнорувати реальну аварію).

Висновок: Оцифровка - це завжди компроміс між зручністю обробки та втратою реальності. Ми перетворюємо живе, неперервне фізичне явище на сухі цифри, з якими легко рахувати, але які водночас роблять систему «сліпою» до певних нюансів і вразливою до маніпуляцій з боку вхідних аналогових сигналів. Таким чином, кібербезпека в КФС починається задовго до написання коду, вона починається на етапі перетворення фізики в цифру. Але ми не можемо відмовитися від оцифровки, бо без неї загальмує розвиток сучасного світу. Саму тому завдання фахівця з кібербезпеки полягає не в

тому, щоб заборонити цифру, а в тому, щоб розуміти слабкі місця на межі між фізикою і цифрою (як-от аліасинг, похибки квантування, незахищені шини даних) і вміти будувати системи, які витримують не лише звичайні технічні збої, а й навмисні маніпуляції.

Крок 2. Нервова система: спілкування всередині системи

Датчики стоять на корпусі корабля чи в моторі, а головний мікроконтролер може знаходитися в іншому відсіку. Для «спілкування» їх між собою існують **шини даних та комунікаційні протоколи** («нервова система» КФС):

- На відміну від звичайного домашнього інтернету, у світі КФС затримка чи втрата певного пакета буде коштувати набагато дорожче. Тому тут використовують спеціалізовані промислові стандарти:

- I2C та SPI: Короткі швидкісні «дроти» для зв'язку чіпів всередині однієї плати.

- CAN-шина (Controller Area Network): Стандарт для автомобілів та рухомих систем. Вона надзвичайно стійка до перешкод (навіть якщо перебити один із дротів, сигнал ітиме по іншому) і дозволяє десяткам пристроїв «спілкуватися» в одній мережі без конфліктів.

- Modbus / Промисловий Ethernet (Profinet тощо): Мережі для обміну даними між контролерами на великих заводах чи електростанціях.

Специфіка промислових стандартів зв'язку (таких як Modbus, CAN, Profibus/Profinet) полягає в тому, що вони проєктувалися зовсім за іншими принципами, ніж звичний офісний чи домашній інтернет. У світі апаратно-програмних комплексів закони фізики й виживання диктують свої правила.

Унікальність цих стандартів полягає в наступному:

1) Жорсткий детермінізм і реальний час.

У домашній мережі, якщо відео на YouTube завантажиться на півсекунди пізніше через затримку, нічого страшного не станеться. У КФС затримка повідомлення може привести до реальної катастрофи. Якщо датчик зіткнення на робот-маніпуляторі надішле команду «зупинитися» із затримкою хоча б у кілька мілісекунд, робот встигне зламатися сам або травмувати людину.

Промислові протоколи гарантують детермінізм: ми точно знаємо, за який максимальний час пакет гарантовано долетить від датчика до контролера. Там немає випадкових затримок чи черг на кшталт звичайного Wi-Fi.

2) Стійкість до агресивного середовища (фізичний рівень)

Заводські цехи, моторні відсіки кораблів чи відкриті електропідстанції - це зони шалених електромагнітних завад (працюють потужні двигуни, зварювальні апарати, висока напруга) та вібрацій. Звичайні мережеві кабелі (вита пара без захисту) там просто «вигорять» від наведень. Промислові стандарти використовують спеціальні фізичні інтерфейси (наприклад,

диференціальні сигнали RS-485 або оптику), які дозволяють передавати дані крізь електромагнітний «шум» без втрати пакетів.

3) *Обмежені ресурси кінцевих пристроїв*

Якщо уявити крихітний датчик температури, запаяний у металеву трубку, то зрозуміло, що у нього немає потужного процесора і гігабайтів оперативної пам'яті. Там стоїть простий 8-розрядний мікроконтролер, який живиться від слабкого струму. Він просто не здатен обробити складний стек протоколів TCP/IP, зашифрувати трафік за допомогою сучасних криптоалгоритмів чи розібрати важкі HTTP-запити. Промислові стандарти (як-от Modbus) побудовані настільки просто, що контролер може прочитати пакети буквально «на льоту», маючи мінімум пам'яті.

4) *Архітектура доступу: жодних «колізій» на лінії*

У звичайному Ethernet, якщо два пристрої одночасно вирішили відправити дані, виникає зіткнення (колізія), пакети б'ються, і мережа на мікросекунду завмирає, повторюючи спробу. Для IT це норма. Для КФС - неприпустимий ризик.

У своєму традиційному вигляді Ethernet проєктувався для передачі великих масивів інформації (файлів, відео, вебсторінок) між комп'ютерами. Ethernet гарантує, що пакет даних дійде до адресата (завдяки протоколам на кшталт TCP, які перевіряють цілісність і просять переслати пакет знову, якщо він загубився). Але він абсолютно не гарантує часу доставки. Якщо пакет затримався на 10 мілісекунд через завантаженість комутатора, для перегляду сайту це непомітно. Ethernet легко масштабується, підтримує тисячі пристроїв в одній мережі. Для роботи з класичним Ethernet потрібні потужні операційні системи, стек TCP/IP, пам'ять і процесори. Маленький датчик температури не може просто так «говорити» по Ethernet.

У кіберфізичних системах (наприклад, у контурі керування швидкісним роботом чи судновим двигуном) концепція «пакет дійде колись там» є неприпустимою. Якщо керуючий контролер відправляє на привод клапана команду «Терміново закритися!», а пакет через чергу в комутаторі затримується на 20 мілісекунд, фізичний процес у цей час уже піде за аварійним сценарієм. У світі операційних технологій потрібен жорсткий детермінізм, треба знати точний час проходження команди до мікросекунди. Класичний Ethernet цього дати не міг через саму свою природу випадкового розподілу трафіку.

Щоб отримати переваги високої швидкості Ethernet, але позбутися його головного недоліку (непередбачуваності затримок), фахівці створили промисловий Ethernet (такі стандарти, як Profinet, EtherCAT, Ethernet/IP, Modbus TCP).

Його специфіка полягає в модифікації на рівні заліза та протоколів:

- *Апаратна синхронізація часу*: усі пристрої в мережі синхронізуються з точністю до наносекунд. Контролер чітко знає, у який момент часу датчик зробив вимір і коли виконавчий механізм отримав команду;
- *Пріоритезація трафіку та обхід черг*: спеціальні комутатори пропускають критичні команди керування поза чергою (механізми

Cut-Through та пріоритетні мітки), блокуючи звичайний інформаційний потік;

- *Адаптація під фізику*: промисловий Ethernet використовує ті самі роз'єми (або спеціальні захищені роз'єми M12), але кабелі та комутатори мають посилений захист від вібрацій, вологи та потужних електромагнітних полів.

Усі ці специфічні особливості (простота, відсутність складного шифрування, довіра до кожного пристрою в межах фізичної лінії) робили промислові стандарти ідеальними для надійності, але катастрофічно вразливими для кібербезпеки. Коли ці закриті шини почали підключати до корпоративних мереж та Інтернету (щоб збирати статистику в хмару), з'ясувалося найголовніше: стандарти проєктувалися для того, щоб вижити в шторм і під час землетрусу, але вони абсолютно не вміють захищатися від зловмисника, який свідомо маніпулює даними на шині.

Коли традиційний IT-світ стикається з промисловими операційними технологіями через Industrial Ethernet, класичний підхід «поставмо антивірус і закриємо двері» не працює. Захист кіберфізичних систем будується зовсім на інших принципах:

1. Жорстке зонування та сегментація: головне правило - IT і OT ніколи не повинні «бачити» одне одного напряму. Архітектура розбивається на рівні (від датчиків на залізі до офісної мережі підприємства). Між ними ставляться буферні зони. Навіть якщо зловмисник зламає офісний комп'ютер бухгалтера, він не буде мати прямого маршруту до промислового контролера.

2. Промислові файрволи з підтримкою DPI (Deep Packet Inspection): звичайний офісний файрвол аналізує лише IP-адреси та порти. Промисловий файрвол значно розумніший, він розуміє специфіку протоколів (Modbus, Profinet тощо). Файрвол перевіряє не просто пакет, а його вміст. Наприклад, він може дозволити команду «читати статус датчика» для будь-кого, але фізично заблокувати команду «змінити прошивку контролера» або «зупинити аварійний клапан», якщо вона надходить із неавторизованої IP-адреси.

3. Дата-діоди (Окремі односторонні ілюзи): якщо керівництву чи хмарним сервісам критично важливо отримувати дані про роботу виробництва чи судна, але виникає ризик зворотного зламу, використовують дата-діоди. Це апаратний пристрій, де фізично розірвані лінії зв'язку для прийому/передачі. Світлодіод світить в один бік (з OT в IT), передаючи потоки телеметрії, але сигнал фізично не може піти назад з IT в OT. Зловмисник може читати дані, але надіслати шкідливу команду в контур управління через цей шлюз неможливо в принципі.

4. Моніторинг аномалій: оскільки промисловий трафік є надзвичайно детермінованим і цикловим (контролер завжди опитує датчики за жорстким розкладом), будь-яке відхилення одразу підозріле. Системи виявлення вторгнень (IDS) аналізують поведінку шини в реальному часі й сигналізують, якщо з'явився незвичайний трафік чи нехарактерна команда.

Захист Industrial Ethernet - це не про софт, а про архітектуру. Ми не намагаємося зробити контролери «абсолютно захищеними» (це неможливо через їхні обмежені ресурси), ми робимо так, щоб зловмисник навіть після зламу IT-периметру фізично не міг дістатися до критичних магістралей керування залізом.

Крок 3. Мозок системи: промислові контролери (PLC)

Коли цифрові дані від усіх датчиків через шини прилітають до «мозку», настає час ухвалення рішень.

У побуті це звичайні крихітні мікроконтролери (як у чайнику чи пральній машині). У промисловості, на суднах чи енергооб'єктах це спеціалізовані PLC (Programmable Logic Controllers) - промислові логічні контролери). Чому не звичайний комп'ютер? Тому що звичайний ПК на Windows може раптово вирішити оновитися чи піти в перезавантаження. Промисловий контролер працює на спеціальних операційних системах реального часу (RTOS), не має зайвих програм, працює роками без вимкнення і гарантовано виконує закладений алгоритм за мілісекунди.

Через свою специфіку (робота без вимкнення, обмежені ресурси, відсутність сучасних механізмів автентифікації у старіших моделях) PLC є найбажанішою цілью для зловмисників.

Основні ризики та кіберзагрози для промислових контролерів:

- *Вразливості в інженерному софті та «логічні бомби»*

Щоб запрограмувати PLC, фахівці використовують спеціалізоване ПЗ (наприклад, Siemens TIA Portal, Rockwell Studio 5000 тощо). Це програмне забезпечення саме по собі стає вектором атаки. Зловмисники можуть заразити комп'ютер фахівця або впровадити приховану «логічну бомбу» (як це було в атаці Stuxnet) безпосередньо в проєктний файл. Коли легітимний фахівець завантажить оновлену програму в контролер, контролер почне виконувати шкідливу інструкцію (наприклад, непомітно розганяти центрифугу до критичних обертів, одночасно показуючи оператору на екрані «все гаразд»).

- *Атаки на доступність*

Промислові контролери мають обмежені обчислювальні ресурси. Якщо завалити PLC шквалом фальшивих або некоректних мережевих запитів, його процесорне завантаження досягне 100%. Контролер просто «зависне» або піде в перезавантаження, скинувши весь контур керування в аварійний (або неконтрольований) стан.

- *Великий «парк» застарілого обладнання (Installed Base)*

Навіть попри появу захищених моделей PLC, на реальних підприємствах, суднах та енергооб'єктах досі масово працюють контролери попередніх поколінь (наприклад, Siemens S7-300/400, старі Modbus-контролери тощо), які проєктувалися в епоху повної довіри і фізично не мають, зокрема, криптографічного захисту. Їхній життєвий цикл становить 20–30 років, і їх ніхто не мінятиме завчасно через високу вартість зупинки виробництва.

- *Людський фактор*

Навіть коли сучасний захищений контролер потрапляє до рук інженерів на місцях, під час пусконаладжувальних робіт або аварійного ремонту вони часто вимикають шифрування та перевірку сертифікатів, залишають дефолтні паролі або відкривають порти «щоб не заважало налаштовувати». Швидкість і зручність обслуговування в операційних технологіях традиційно перемагають безпеку.

- *Вразливості в самій реалізації захисту.*

Наявність криптографії чи автентифікації не гарантує захищеності. Дослідники безпеки регулярно знаходять у прошивках сучасних PLC вразливості нульового дня (0-day), обходи криптографічних перевірок або слабкі місця в реалізації стека безпеки самого вендора.

Таким чином, сучасні PLC часто мають засоби автентифікації на папері та в архітектурі, але на практиці їхній захист часто нівелюється або застарілим парком обладнання, або людським фактором при експлуатації.

На відміну від звичайного сервера, де зламаний хост просто втрачає дані чи не працює сайт, злом PLC напряду перетворюється на фізичні наслідки у реальному світі: згорілі мотори, розірвані труби, аварії на електростанціях чи втрата керування судном. Саме тому безпека PLC вимагає не просто захисту периметра, а жорсткого контролю цілісності коду та апаратної ізоляції.

Крок 4. Зворотний шлях

Мозок ухвалив рішення: *«Треба терміново повернути клапан подачі палива на 15 градусів!»*. Тут починається рух у зворотний бік:

1. ЦАП (Цифро-аналоговий перетворювач): переводять цифрову команду назад в електричний аналоговий сигнал або потужні імпульси живлення.

2. Драйвери та реле: підсилюють слабкий сигнал, подаючи на нього високу напругу чи великий струм від силової мережі.

3. Виконавчі механізми (актуатори): електродвигуни, соленоїди, гідравлічні циліндри чи пневмоклапани отримують енергію і виконують реальну механічну роботу - зрушують стерно, перекривають трубу чи гальмують колесо.

Кібератаки на КФС часто відбуваються на рівні заліза та його протоколів! Зловмиснику не обов'язково ламати складний софт, він може надіслати підроблену цифрову команду через шину даних (скажімо, сфальсифікувати показник датчика температури або змусити контролер повірити, що тиск у котлі нормальний, хоча він критичний).

Будь-який технічний розвиток кіберфізичних систем — підвищення швидкості оцифровки, розширення пропускну здатності Industrial Ethernet, інтеграція автономних компонентів чи перехід до складніших архітектур — не вирішує проблем безпеки автоматично, а лише розширює простір потенційних загроз. У сучасних КФС еволюція технологій і розвиток кібербезпеки є невіддільними: що складнішою, інтегрованішою та гнучкішою стає система керування фізичним світом, то критичнішою стає потреба в архітектурному захисті кожного її рівня.

Питання для самоперевірки:

1. Визначення кіберфізичної системи у відповідності з NIST.
2. Поясніть твердження: «Кіберфізична система - це трикутник, де обчислення, мережі зв'язку та фізичні процеси утворюють єдиний організм».
3. Наведіть приклади КФС. Поясніть, чому саме наведену Вами систему можна віднести до КФС.
4. В чому полягають фундаментальні причини, що призвели до розвитку КФС?
5. Як Ви можете пояснити, що «ефективність, швидкість і автономність завжди воюють із безпекою»?
6. Три обов'язкові фундаментальні властивості КФС. Охарактеризуйте кожную з них.
7. Який наслідок має будь-який кібер-інцидент для КФС? Поясніть свою відповідь.
8. Перерахуйте основні сфери використання КФС.
9. Перерахуйте і поясніть відмінності між ІТ-системами та КФС.
10. Поясніть ієрархію пріоритетів безпеки (так звана тріада CIA) в КФС.
11. Що є головним пріоритетом в КФС? Поясніть свою відповідь.
12. Які наслідки в сенсі кібербезпеки КФС має оцифровка використовуваних системою аналогових сигналів?
13. Поясніть, чому з погляду кібербезпеки аліасінг виступає як зброя зловмисника.
14. В чому полягає специфіка промислових стандартів зв'язку?
15. На яких загальних принципах будується захист кіберфізичних систем? Пояснити кожний з цих принципів.
16. Основні ризики та кіберзагрози для промислових контролерів.

Тема 2. ЦІЛЬОВІ АТАКИ НА КІБЕРФІЗИЧНІ СИСТЕМИ

План

1. Зміна парадигми: від викрадення даних до деструктивного фізичного впливу
2. Багатоступеневі сценарії проникнення в кіберфізичні системи
3. Знакові інциденти
4. Постатаковий ефект та етапи маскуванню

1. Зміна парадигми: від викрадення даних до деструктивного фізичного впливу

Як вже обговорювалося в попередній темі, класичні кібератаки в ІТ спрямовані на конфіденційність (крадіжка паролів, вимагання грошей через шифрування баз даних тощо). У кіберфізичних системах метою атак є цілісність і доступність процесу, що може призвести до катастрофічних наслідків у реальному фізичному світі (руйнування обладнання, екологічні аварії, загроза життю людей).

1. В ІТ-системі цілісність означає, що байти в базі даних не змінені. У КФС дані самі по собі вторинні. Справжня мета деструктивної атаки - вивести керований об'єкт за межі фізичних інваріантів (допустимих діапазонів тиску, температури, швидкості обертання, напруги). Треба зазначити, що у кожного пристрою у фізичному світі є свої межі безпеки - інваріанти. Наприклад, температура в реакторі не має перевищувати 300 градусів, а тиск у трубі 10 атмосфер. Якщо перетнути цю межу, метал не витримає, і станеться аварія. Звичайний хакер в інтернеті змінює бази даних. Хакер у кіберфізичній системі намагається зламати саму фізику процесу - обійти захисні датчики і примусити обладнання працювати в завідомо аварійному режимі, поки воно фізично не зламається.

Найбільш специфічними для КФС є таргетовані атаки.

Таргетована атака - це цілеспрямована, ретельно спланована кібератака, спрямована проти конкретної організації, підприємства, інфраструктури або навіть однієї конкретної людини.

Якщо порівнювати звичайні масові кібератаки та таргетовані, різниця дуже добре помітна:

- *Масова атака (як риболовля сіткою)*: шахраї розсилають мільйони однакових фішингових листів або сканують інтернет у пошуках випадкових «дірок» у захисті. Хто попався, той попався. Їм байдуже, чий це сервер - невеликого інтернет-магазину чи великого заводу. Головне - це зібрати хоч якийсь «улов» (гроші з карток, легкі паролі).
- *Таргетована атака (як полювання за конкретним звіром)*: це спецоперація. Зловмисники заздалегідь чітко знають, кого саме вони хочуть зламати (наприклад, конкретну електростанцію, завод з виробництва деталей, судно певного типу чи систему управління транспортом).

Головні риси таргетованої атаки:

- Зловмисники довгий час *готуються*, вивчають свою жертву: які комп'ютери там стоять, якими програмами користуються інженери, хто з працівників який ланцюжок пошти читає, де є слабкі місця в захисті;
- Під таку атаку часто пишуть *спеціальні* шкідливі програми під конкретне обладнання (як це було у випадку зі **Stuxnet**, який шукав цілком конкретні контролери на урановому заводі). Звичайний антивірус такі віруси зазвичай не бачить, бо їх створено в єдиному екземплярі.

Проникнувши в мережу, хакери можуть місяцями сидіти там «тихіше води», не видаючи себе, вивчаючи процеси зсередини, і чекати ідеального моменту для удару.

У контексті кіберфізичних систем таргетовані атаки - це інструмент професіоналів або спецслужб, адже щоб зупинити завод чи нашкодити кораблю, шаблонні віруси з інтернету не спрацюють. Потрібен заздалегідь підготовлений розрахунок.

Зауважимо, що зловмисник може навіть не намагатися безпосередньо зламати алгоритм роботи датчика чи вимкнути його. Головна майстерність таргетованої атаки полягає в тому, щоб створити стан, коли контур управління розривається на рівні сприйняття: фізичний об'єкт уже руйнується (наприклад, критично зростає тиск у реакторі), але математична модель, яку бачить диспетчер, залишається абсолютно «зеленою» і стабільною завдяки підміні потоків даних на рівні пам'яті контролера.

- **Асиметрія часу.** Швидкість протекання фізичних процесів у кіберфізичній системі (теплообмін, гідроудар, механічні вібрації) значно вища за швидкість реакції людини-оператора чи навіть час оновлення стандартних опитувань мережі. Зловмисники розраховують сценарії так, щоб критичний імпульс руйнування стався впродовж кількох секунд, коли система ще вважається «безпечною».

Атаки на КФС зазвичай є державно-спонсорованими і вимагають тривалої розвідки, глибокого розуміння технологічного процесу та наявності специфічного тестового обладнання.

2. Багатоступеневі сценарії проникнення в кіберфізичні системи

Кіберфізичні системи рідко підключені до прямого інтернету, тому вектор проникнення вимагає складних багатоступеневих сценаріїв.

У кібербезпеці кіберфізичних систем є золоте правило: жоден серйозний об'єкт неможливо зламувати «в лоб». Не можна просто сісти за ноутбук і підключитися до заводського контролера чи системи управління судном через інтернет. Вони фізично ізольовані. Саму тому хакери (особливо державні угруповання) будують довгий, багатоступеневий ланцюжок проникнення, який може тривати місяцями. Розглянемо цей шлях покроково.

Крок 1. Атака на корпоративне IT та підрядників

У будь-якого секретного заводу, електростанції чи судноплавної компанії є офіс. Там бухгалтерія купує папір, менеджерки п'ють каву, а директор підписує договори. І головне - у цього офісу є звичайний інтернет і корпоративна пошта. Підприємства ніколи не працюють на самоті: на них за

контрактом працюють десятки сторонніх підрядників (хтось налагоджує кондиціонери, хтось ставить софт, хтось постачає деталі). Хакери починають з фішингу. Наприклад, надсилають головному інженеру або менеджеру підрядника гарно оформлений лист від імені міністерства чи відомого вендора обладнання. Усередині - заражений документ Word або запаролений архів. Людина відкриває файл, і шкідливий код непомітно осідає на офісному комп'ютері. Корпоративна мережа підприємства зламана. Але це ще *не* промислова мережа, це лише трамплін.

Крок 2. Розвідка на місці та горизонтальне переміщення

Опинившись усередині офісної мережі (ІТ), хакери не квапляться атакувати виробництво. Вони починають «тиху гонитву», місяцями вивчають обстановку: шукають, де лежать паролі до адміністративних облікових записів, вивчають внутрішню документацію, схеми мереж, інструкції для інженерів; шукають мости чи шлюзи, які з'єднують офісний світ (ІТ) із промисловим світом (ОТ). За правилами безпеки ці мережі мають бути жорстко розділені, але через людський фактор або «потрібно для зручності» там майже завжди знаходяться дірки, наприклад, ноутбук якогось системного адміністратора, який підключається і туди, і туди.

Крок 3. Долання «повітряного зазору»

Коли шлюз знайдено, перед хакерами постає стіна - найбезпечніша ділянка заводу чи судна, де крутяться самі контролери, повністю відрізані від зовнішнього світу. Мережею туди не зайти. Що робити? Тут у хід ідуть найхитромудріші трюки. Хакери заражають флешки, якими користуються інженери під час перевірки обладнання. Інженер вставляє флешку в офісний ноутбук (де є інтернет і вірус), а потім приносить її в «чисту» зону й підключає до пульта керування станцією чи судном. Вірус потрапляє всередину. Або: зловмисники зламують компанію, яка пише легітимне програмне забезпечення для цих контролерів. Коли інженери заводу скачують чергове «офіційне та безпечне» оновлення софту від улюбленого вендора, разом із ним вони встановлюють у систему прихованого трояна. Виникає питання: і що ж робити?

Якщо фізичний розрив у реальному світі все одно можна обійти за допомогою людського фактора чи ланцюжка постачання, то будувати захист на принципі «ми повністю відрізані від світу, значить, ми невразливі» - це шлях до катастрофи. Захиститися на «Кроці 3», коли загроза вже намагається перетнути межу між офісом і виробництвом або потрапити через носії, допомагає цілий комплекс сучасних інженерних та організаційних рішень:

3.1. Повна відмова від «абсолютної довіри» всередині (Zero Trust в ОТ)

Раніше думали, що якщо ти вже зайшов у промислову мережу, то значить ти «свій» (інженер, оператор). Але сьогодні це не ефективно. Сьогодні впроваджують архітектуру *Zero Trust*. Навіть якщо шкідливий код уже потрапив на контролер через флешку, він не зможе просто так надіслати команду на зупинку обладнання. Контролер вимагає багаторівневої автентифікації, криптографічних підписів команд або підтвердження від інших незалежних систем безпеки.

3.2. *Захищені станції оновлення та передачі даних*

Інженерам усе одно треба переносити проекти, оновлювати прошивки чи збирати логи з «чистої» зони. Але робити це зі звичайних флешок тепер заборонено. Створюються спеціальні шлюзи безпеки (так звані *Secured Jump Hosts* або термінали оновлення). Перш ніж флешка чи ноутбук інженера потраплять у промислову мережу, вони проходять через суворий ізольований комп'ютер-сканер, який перевіряє файли на віруси, аналізує цифрові підписи і блокує будь-який сторонній код чи несанкціоновані виконувані файли.

3.3. *Контроль ланцюжка постачання*

Оскільки оновлення софту від вендорів часто стають «троянським конем», правила гри тут стають жорсткішими: промислові підприємства більше не вірять оновленням на слово. Вони вимагають від виробників заліза криптографічно підписаних пакетів оновлень, проходять власний аудит безпеки коду перед тим, як залити його на контролери, і тримають тестові стенди («цифрові двійники»), на яких спочатку ганяють нові прошивки, перевіряючи їх на дивну поведінку.

Цифрові двійники (Digital Twins) зараз одна з найбільш гарячих і водночас практичних тем у світі промисловості та кіберфізичних систем. Морський лайнер, складний завод чи сучасний безпілотний комплекс - це надзвичайно складні механізми, де кожна помилка коштує мільйони, а будь-який збій може призвести до аварії. Замість того, щоб тестувати нові команди, оновлення чи аварійні сценарії на «живому» залізі, створюється його віртуальна копія в комп'ютері. Цифровий двійник - це віртуальний двійник реального фізичного об'єкта, який живе з ним «паралельним життям» у режимі реального часу. Між ними постійно триває двосторонній обмін інформацією:

- *Знизу вгору (від фізичного до цифрового):* тисячі датчиків на реальному судні чи заводі збирають дані про вібрацію, температуру, тиск, швидкість і струм і передають їх у віртуальну модель. Двійник завжди знає, в якому стані його «живий брат»;
- *Згори вниз (від цифрового до фізичного):* інженери можуть проводити на віртуальній моделі будь-які експерименти, наприклад, «а що буде, якщо різко вивернути стерно під час шторму на швидкості 30 вузлів?» або «як поведе себе двигун, якщо нагріти його до 400 градусів?». Усі ці розрахунки робляться у віртуальному світі. Якщо модель каже, що все гаразд, висновки чи оновлення безпечно застосовують на реальному об'єкті.

Якщо раніше інженери проєктували системи на папері або в простих програмах, запускали їх у реальному світі, а потім латали дірки під час аварій, то сьогодні такий підхід занадто дорогий і небезпечний. Цифрові двійники дають революційні переваги. По-перше, віртуальна модель знає, як зношуються деталі. Двійник може просигналізувати: «У цього насоса на реальному судні залишилося приблизно 50 годин роботи, хоча зовні він виглядає нормально, краще замінити його зараз». Це рятує від аварій. По-друге, не можна влаштувати справжню аварію на атомній станції чи нафтовій

платформі, щоб перевірити, як спрацює захист. Але можна симулювати найгірший сценарій у цифровому двійнику мільйон разів і подивитися, де виникають слабкі місця. По-третє, перш ніж залити нову прошивку в контролери реального підприємства чи безпілотної, її спочатку «проганяють» на цифровому двійнику. Якщо хакери спробують закинути туди шкідливий код, симуляція покаже аномальну поведінку системи ще до того, як зловмисники завдадуть шкоди реальному залізу.

Сьогодні цифрові двійники перейшли з категорії «технологій майбутнього» у статус промислового стандарту для критичних галузей:

- *Аерокосмічна галузь та авіація*: NASA і провідні авіабудівні компанії давно створюють повні цифрові копії космічних апаратів і літаків. Якщо на орбіті виникає проблема, на Землі в лабораторії відтворюють точно таку ж ситуацію на цифровому двійнику і шукають вихід;
- *Морський транспорт і MASS*: сучасні автономні судна будують спочатку часто як цифровий двійник. Усі алгоритми штучного інтелекту, навігації та уникнення зіткнень мільйони годин «тренуються» у віртуальному океані, перш ніж їх встановлять на реальний корпус корабля;
- *Промисловість*: великі заводи та енергетичні компанії створюють цифрові копії своїх цехів, щоб оптимізувати потоки енергії, прогнозувати затори на виробництві та блокувати несанкціоновані втручання в мережі.

Таким чином, цифровий двійник для кіберфізичної системи - це одночасно і суперсила (бо дозволяє бачити невидиме, передбачати аварії та тестувати системи), і нова ціль для хакерів. Якщо хакеру важко пробити захист безпосередньо на заводі чи на борту автономного судна (бо там стоять апаратні замки і жорсткі фільтри), він може спробувати атакувати цифрового двійника. Основні вектори таких атак:

- *Отруєння даних*: цифровий двійник живе завдяки потоку даних від датчиків реального об'єкта. Якщо зловмисник перехопить і підробить цей потік на шляху від датчиків до віртуальної моделі (наприклад, змусить двійник вірити, що тиск у нормі, хоча він уже росте), двійник почне давати інженерам хибні прогнози.
- *Атака на алгоритми симуляції*: хакери проникають у хмару або сервер, де крутиться сам цифровий двійник, і тихо змінюють його математичну модель чи фізичні коефіцієнти. Коли фахівці запусають на цьому двійнику тестування нового автопілота, він видасть хибні висновки, і на реальне судно завантажать небезпечне оновлення;
- *Злам каналу зворотного зв'язку*: коли цифровий двійник розраховує «ідеальну» оптимізацію або оновлення і надсилає його назад на реальний завод, зловмисник може перехопити цей пакет і підмінити його.

Оскільки цифровий двійник - це критична ланка керування, його безпека будується за найсуворішими канонами:

- *Криптографічний захист*: кожен пакет даних, який надходить від датчиків корабля до цифрового двійника і назад, підписується криптографічними ключами. Якщо дані хоча б на біт змінені - система їх повинна відкинути;
- *Ізольовані середовища та конфігураційний контроль*: сама модель двійника зберігається в захищеному резерві. Її код регулярно перевіряється на несанкціоновані зміни за допомогою хеш-тестів;
- *Перехресна перевірка*: сучасні системи порівнюють прогнози цифрового двійника з тим, що реально відбувається у фізичному світі в цю ж секунду. Якщо віртуальна модель каже одне, а датчики показують інше, це є сигналом про атаку, що відбулася на цифровий двійник.

Чи можна сьогодні працювати без цифрового двійника? **У сучасних високотехнологічних галузях не можна.** Сучасні кіберфізичні системи (наприклад, автономні судна MASS або складні безпілотні авіаційні комплекси) настільки складні, що людський розум уже не здатен утримати в голові всі взаємозв'язки їхніх датчиків, алгоритмів ШІ та механіки. Автономний корабель має «проплисти» у віртуальному океані мільйони віртуальних кілометрів, перш ніж випустити його у відкрите море. На реальні випробування пішли б десятиліття.

Без повноцінних цифрових двійників ще працюють простіші, консервативні виробництва (наприклад, невеликі заводи з випуску стандартних деталей), де ціна помилки низька, а процеси не змінюються роками. Але що складніша й дорожча система, то критичнішою є наявність її цифрової копії.

Таким чином, захист зв'язку між фізичним світом, цифровим двійником і контуром управління - це передній край сучасної кібербезпеки.

3.4. Апаратні обмеження

На критичних контролерах фізично блокують порти (наприклад, USB-роз'єми заливають епоксидною смолою або закривають металевими заглушками із замками, якщо вони не потрібні для щоденної роботи); використовують апаратні модулі захисту (наприклад, спеціальні перемикачі на платах контролерів у режим «тільки читання»), коли під час штатної роботи змінити прошивку фізично неможливо, для цього інженер має особисто прийти до шафки з контролером і перемкнути тумблер фізично.

3.5. Однонаправлені шлюзи - головна зброя проти мережевого переходу

Якщо дані з виробництва треба витягнути нагору (наприклад, для аналізу начальством у офісі), але при цьому не можна допустити, щоб з офісу в ОТ-контур пройшов хоч один зайвий пакет, використовують однонаправлені шлюзи - фізичні пристрої (на рівні оптоволокна), які дозволяють світлу (і даним) текти тільки в один бік - з виробництва назовні. Хакер може скільки завгодно зламувати офісну мережу, але через дата-діод назад у бік контролерів сигнал просто не пройде.

Таким чином, повітряний зазор - це ілюзія безпеки, якщо за ним не стоять жорсткі правила інженерної дисципліни, криптографічний контроль коду та апаратні обмеження.

Крок 4. Закріплення та маскування в ОТ-контурі

Коли шкідливий код нарешті опинився всередині контролера або промислового комп'ютера, він робить головне: ховається від антивірусів. Промислові контролери працюють на специфічних операційних системах реального часу, які не мають звичних нам захисних програм чи екранів моніторингу. Вірус прописується глибоко в пам'яті заліза (перепишує прошивку або мікрокод), стаючи її частиною. Тепер, навіть якщо ви перезавантажите систему чи зміните операційну систему на комп'ютері, вірус усе одно залишиться всередині контролера. Якщо шкідливий код «прошився» на рівні мікросхеми (записався у флеш-пам'ять або завантажувач мікроконтролера), звичайна перевстановлення Windows, форматування чи заміна жорсткого диска на комп'ютері не допоможуть. Для контролера цей вірус уже став «рідною» частиною його власної операційної системи. Але з цим можна боротися покроково:

4.1. Апаратний захист від запису

Найкращий спосіб захистити прошивку контролера - фізично заблокувати можливість її зміни. Багато промислових контролерів мають спеціальний фізичний перемикач на платі (тумблер). Коли система налаштована й працює на заводі чи на судні, фахівець фізично перемикає тумблер у положення «Тільки читання». Після цього вірус фізично не зможе перезаписати мікрокод чи прошивку - мікросхема заблокує будь-яку спробу запису на апаратному рівні. Щоб оновити прошивку, треба фізично прийти до шафи керування і перемкнути тумблер наживо.

4.2. Захищене завантаження

Сучасні промислові контролери переймають технології, які давно працюють у сучасних комп'ютерах та смартфонах. Кожного разу, коли контролер умикається або перезавантажується, він не починає одразу крутити двигуни чи відкривати клапани. Спочатку спрацьовує спеціальний незмінний код усередині чіпа (Secure Boot). Цей код перевіряє цифровий підпис прошивки за допомогою зашифрованих математичних ключів, які заклав туди виробник. Якщо підпис справжній і не змінений жодним хакером, система стартує. Якщо ж у коді прошивки змінено хоча б один байт, контролер розуміє, що його зламали, блокує запуск і подає сигнал тривоги.

4.3. Криптографічний контроль цілісності та Audit Logs

Якщо повна заміна прошивки неможлива без фізичного доступу, зловмисники намагаються хоча б непомітно правити параметри конфігурації в пам'яті контролера. Як це виявляють? Системи моніторингу постійно порівнюють поточну контрольну суму (хеш) пам'яті контролера з еталонною, яка зберігається в захищеному місці. Якщо суми не збігаються - система одразу сигналізує, що в контролер хтось «лазив».

4.4. Довірене «залізо» та заміна чіпів

А що робити, якщо контролер вже заражений на рівні прошивки, і апаратного захисту не було? У такому разі простим перепрошиванням через мережу часом уже не обійтися (адже вірус міг заблокувати і саму процедуру оновлення). Проводиться так звана апаратна реанімація: виймають плату контролера, підключають її на спеціальному заводському стенді через спеціальні сервісні порти, повністю стирають усю пам'ять на спеціальному програматорі та заливають «чистий» заводський образ системи. А в особливо важких випадках просто фізично викидають заражений мікроконтролер і ставлять новий.

Таким чином, у світі ІТ зламаний комп'ютер можна перевстановити за 15 хвилин. У світі КФС, якщо заражено «мозок» заліза на рівні прошивки, ремонт може вимагати фізичного демонтажу обладнання. Тому головна задача безпеки - не допустити первинного запису, використовуючи апаратні перемикачі захисту та Secure Boot.

Треба зауважити, що сучасна кібербезпека на виробництві чи на транспорті часто працює за принципом «острівної оборони»: компанії впевнені, що *«якщо в нас немає інтернету в цеху, значить, ми в безпеці»*. Багато хто витрачає мільйони на залізні двері, охорону з автоматів та фізичні огляди, забуваючи, що зловмисник може проповзти крізь цей «захисний периметр» через звичайну флешку інженера, який просто хотів перенести робочий кресленик. Ось чому в КФС безпека буде зовсім інакше.

3. Знакові інциденти

3.1. Вірус Stuxnet

Головною мішенню вірусу став завод по збагаченню урану в Ірані. Усе почалося наприкінці 2000-х років. Тоді Іран активно намагався збагачувати уран на ядерному об'єкті в місті Натанз. Для цього використовувалися тисячі центрифуг, щоб розділяти ізотопи. Цей завод був справжньою фортецею: він не був підключений до інтернету, вхід туди охоронявся краще, ніж сховище Центрального банку. Проте уряди кількох країн, **зокрема США** вирішили зупинити цю програму без оголошення відкритої війни.

Оскільки в інтернеті завод був недоступний, хакерам потрібно було якось занести туди вірус фізично. І вони використали людський фактор. Stuxnet спочатку поширювався по всьому світу через звичайні носії, зокрема флешки. Він заражав звичайні офісні комп'ютери, але не робив нічого поганого, він просто тихо «сидів» у пам'яті й чекав. Вірус «знав», що він потрапив куди треба, лише тоді, коли комп'ютер підключався до специфічних промислових контролерів *Siemens*, які використовувалися саме на ядерних заводах Ірану. Коли хтось із інженерів привіз заражену флешку всередину захищеної зони Натанза і вставив у робочий комп'ютер, вірус нарешті дістався до цілі.

Коли Stuxnet опинився всередині закритої мережі заводу, він знайшов головний комп'ютер управління і спеціальні контролери, які керували швидкістю обертання центрифуг. Час від часу (наприклад, раз на тиждень на п'ятнадцять хвилин) вірус змінював команду і розганяв центрифуги до

божевільної швидкості, а потім різко гальмував. Для крихких роторів центрифуг такі стрибки навантаження були смертельними, метал починав кристалізуватися, ламатися і рватися зсередини. Центрифуги масово виходили з ладу.

Найстрашніше у Stuxnet було навіть не те, як він ламав залізо, а те, як він це приховував. Якщо на заводі починає рватися обладнання, зазвичай усе має заверещати, а на екранах з'явиться червона тривога. Але Stuxnet зробив інакше. За секунду до того, як розігнати центрифуги, вірус перехоплював управління екранами і запускав у пам'ять комп'ютера заздалегідь записане відео або старі показники датчиків. Оператор сидів перед монітором і бачив спокійні зелені графіки, які визначали ситуацію, як нормальну. А в цей час металеві циліндри розліталися на шматки. Іранські інженери довгий час не могли зрозуміти, чому їхнє обладнання ламається саме по собі.

Вірус Stuxnet увійшов в історію як перша у світі цифрова (кібер) зброя. До Stuxnet хакери крали гроші чи ламали сайти. Stuxnet став першим вірусом, який завдав реальних, фізичних руйнувань у матеріальному світі. Цей вірус був настільки «розумним», що ігнорував будь-які інші комп'ютери у світі й шукав лише конкретну «пастку» на конкретному заводі, не зачіпаючи випадкових людей. Саме після історії зі Stuxnet уряди, заводи та військові всього світу зрозуміли: кібербезпека - це не просто захист від спаму в пошті. Це питання фізичного виживання критичної інфраструктури. Така кібернетична історія ідеально ілюструє, чому захист кіберфізичних систем є питанням державної ваги.

3.2. *CrashOverride / Industroyer (Атаки на енергосистему України)*

Перший приклад шкідливого ПЗ, спроектованого як універсальна платформа для прямого нападу на енергетичну інфраструктуру. Якщо Stuxnet був розрахованим на один конкретний завод, то CrashOverride — це вже кіберзброя масового ураження для енергосистем.

Саме ця атака увійшла в історію як перший випадок, коли хакери використали спеціалізований вірус, щоб напряду вимкнути світло в цілому місті чи регіоні.

У грудні 2015 року в Україні вперше у світі сталася масова кібератака на енергопостачання (тоді постраждав «Прикарпаттяобленерго» та інші компанії), коли десятки тисяч людей раптово залишилися без світла на кілька годин. А вже у 2016 році з'явився шкідливий код CrashOverride (або Industroyer), який був створений саме як універсальний «робот-знищувач» електромереж.

Звичайні віруси (наприклад, віруси-вимагачі) шифрують файли на комп'ютері і вимагають гроші. Але енергосистема не працює на звичайних текстових файлах чи базах даних. На підстанціях усе контролюється спеціальними промисловими контролерами, які керують величезними вимикачами, трансформаторами та релейним захистом. І ці пристрої говорять не людською мовою, а спеціальними промисловими протоколами (IEC 60870-5-104, IEC 61850, OPC тощо).

Розробники CrashOverride зашили в код цього вірусу пряме розуміння електричних протоколів. Вірус не просто ламав операційну систему комп'ютера диспетчера. Він усередині себе ніби мав «словарі» та інструкції для підстанцій. Потрапивши в мережу, він починав пряму надсилати командам підстанцій команду «Вимкнути живлення».

Зсередини ця атака виглядала наступним чином. Спочатку вірус місяцями тихо «сидів» усередині мережі компанії, вивчаючи схеми, де які трансформатори підключені, і чекаючи команди. У призначену годину вірус активувався. Він одночасно надсилав команди на десятки автоматичних вимикачів на підстанціях. Величезні масиви обладнання синхронно розмикали контакти. Місто чи район миттєво занурювалися в темряву. Але найстрашніше було далі - *ефект подвійного удару*.

Хакери розуміли, що просто вимкнути світло - це лише півсправи. Електрики приїдуть, перезавантажать систему, увімкнуть рубильники назад, і все запрацює за кілька годин. Тому CrashOverride мав у собі ще й руйнівний модуль: перед тим як залишити систему, вірус спеціально переписував внутрішню мікропрограму (завантажувачі) деяких захисних контролерів, роблячи їх повністю «цегляними». Це означало, що після відключення світла диспетчери не могли просто так натиснути кнопку відновлення, контролери виявилися спаленими на рівні «мізків», і їх довелося міняти фізично. Це затягнуло відновлення роботи набагато довше.

CrashOverride змінив правила гри для всього світу. Це був перший конструктор атак. Розробники могли легко додавати до нього «модулі» для інших країн чи інших протоколів, просто дописуючи нові команди зв'язку. Вірус показав, що цілком тепер стає не комп'ютер бухгалтера, а фізична стійкість цілої держави - світло, тепло, вода на насосних станціях. Після цього випадку всі енергокомпанії світу почали масово впроваджувати системи виявлення аномалій у трафіку (той самий промисловий DPI), щоб бачити, якщо хтось намагається надіслати команду «вимкнути рубильник» у неробочий час або з неавторизованого місця.

3.3. *TRITON – атака на систему протиаварійного захисту*

Якщо Stuxnet був зброєю проти ядерних центрифуг, а CrashOverride — проти енергомереж, то одним з найбільш лякаючих прикладів із сучаснішої історії є атака на систему протиаварійного захисту, відома як TRITON (або TRISIS), що сталася у 2017 році. Це еволюційний стрибок у світі кібердиверсій, бо хакери атакували не просто управління, а «останню лінію оборони», яка мала захищати людей і обладнання.

Якщо станеться аварія на величезному нафтохімічному заводі або нафтопереробному комбінаті (наприклад, тиск підскочить до критичної позначки, а звичайний комп'ютер зависне), це може привести до величезних негативних наслідків, аж до загибелі людей. Щоб такого не сталось, на таких об'єктах стоїть окрема, незалежна система протиаварійного захисту (Safety Instrumented System, SIS). Це «червона кнопка» всього заводу: її єдина задача - у разі загрози вибуху миттєво знеструмити все обладнання і скинути тиск. Вона працює на абсолютно іншому «залізі», повністю відрізаному від

робочих екранів операторів. Розробники вірили, що навіть якщо весь завод зламають хакери, система безпеки увімкнеться і врятує людей.

У 2017 році на критичному інфраструктурному об'єкті на Близькому Сході хакери зуміли пробити цей останній бастіон. Вони створили шкідливий код TRITON, розроблений спеціально для атак на контролери безпеки компанії Schneider Electric (Triconex). Хакери не просто хотіли зупинити завод. Їхня мета була набагато страшнішою: вимкнути саму систему безпеки, зробити її сліпою і глухою. План полягав у тому, щоб після вимкнення захисту штучно створити аварійну ситуацію на виробництві (наприклад, підняти тиск у реакторі до загрози вибуху), при цьому система захисту не мала спрацювати, бо її попередньо «відключили» зсередини вірусом. Підсумок мав бути катастрофічним - реальний вибух, викид токсичних речовин і людські жертви. Це була кібератака, спрямована на пряме фізичне знищення людей та інфраструктури, а не просто на припинення роботи підприємства.

Світ дивом уникнув масштабної техногенної катастрофи завдяки одній дрібниці: хакери припустилися помилки у своєму шкідливому коді. Коли вірус намагався переписати пам'ять контролера безпеки й вимкнути захист, він випадково зробив дещо не так, як очікувалося. Промисловий контролер безпеки за своєю архітектурою, як правило, спроектований так, що якщо з пам'яттю або програмою коїться щось підозріле, контролер не мовчить, він автоматично переходить у безпечний режим і повністю зупиняє роботу всього заводу. Саме це і сталося. Якби не захисна логіка самого «заліза», яка спрацювала проти планів хакерів, наслідки могли бути жахливими.

TRITON довів, що сучасні кібератаки на КФС ціляться не тільки на гроші чи дані, а на фізичне виживання людей і руйнування матеріального світу; навіть найізольованіші системи безпеки можуть бути зламані. Тепер такі системи проєктуються з урахуванням того, що ворог уже всередині, і вимагають фізичного підтвердження будь-яких змін через фізичні ключі та багаторівневу апаратну ізоляцію.

4. Постатаковий ефект та етапи маскування

Коли атака вже відбулася, для її авторів починається дуже важлива фаза. Їхня мета тепер - не просто нашкодити, а зробити так, щоб ніхто не зрозумів, хто, як і чому це зробив, а також максимально ускладнити відновлення системи.

Хакери чудово розуміють, що після аварії на місце прибудуть спеціалісти з кіберкриміналістики (Digital Forensics). Вони почнуть досліджувати жорсткі диски, логі та пам'ять контролерів. Що роблять зловмисники, щоб замести сліди? Шкідливий код самотужки вичищає всі системні журнали (Event Logs), куди операційна система чи програми записували свої дії; хакери можуть змінити дати створення та модифікації шкідливих файлів на контролерах чи серверах, щоб вони виглядала так, ніби вони «лежали тут роками» ще з моменту заводського збирання, збиваючи з пантелику слідчих.

Вимкнути обладнання за допомогою софту - це занадто просто. Справжні таргетовані атаки часто розраховані на те, щоб фізично заблокувати можливість швидкого відновлення системи. Для цього:

- вірус може спеціально переписати мікропрограму нижнього рівня (наприклад, флеш-пам'ять материнської плати контролера або захисного реле) сміттєвими даними;
- після такого перезавантаження мікроконтролер стає просто шматком кремнію та пластику («цеглою»). У нього повністю стертий «мозок», і він більше ніколи не ввімкнеться самостійно.

Як результат, замість того, щоб просто натиснути кнопку «Перезавантажити» на віддаленому комп'ютері, аварійним бригадам часто доводиться ліквідувати наслідки аварії впродовж днів або навіть тижнів.

Для організаторів атаки дуже важливим серед іншого є психологічний ефект від неї. Коли система «падає», на екранах диспетчерів може починатися справжній хаос: вистрибують сотні помилок одночасно, міняються індикатори, зникає зв'язок з половиною датчиків. Це робиться навмисно, щоб дезорієнтувати черговий персонал. Поки інженери витрачають дорогоцінні хвилини на те, щоб просто зрозуміти, де саме зламалося і що взагалі відбувається, час для запобігання катастрофі втрачається.

Основним правилом фахівця з кібербезпеки тут є наступне: *не можна довіряти жодному пристрою після аварії*. Саму тому на сучасних критичних об'єктах створюють так звані незмінні резервні копії систем, які зберігаються в закритому офлайн-середовищі на фізичному носії так, щоб хакер під час атаки ніяк не зміг туди дотягнутися. Якщо залізо перетворилося на «цеглу», його не намагаються лікувати, його просто замінюють на чистий модуль з еталонного зібрання.

Таким чином, сучасний кіберзахист КФС - це вже давно не захист від «вірусів», це боротьба з інтелектуальним супротивником у **фізичному** просторі.

Фахівці з кібербезпеки мають мислити не категоріями «чи є у нас антивірус», а категоріями **фізичних** інваріантів, перевірки цілісності заліза та постійного аудиту реального стану речей у світі.

Питання

1. Що є головним об'єктом атаки у кіберфізичних системах на відміну від класичного ІТ? Чому порушення конфіденційності там часто відходить на другий план?
2. Що таке «фізичні інваріанти» процесу і чому їхнє порушення є справжньою ціллю деструктивної кібератаки на КФС?
3. Опишіть поняття «сліпої зони управління» на прикладі операційного центру керування об'єктом. Яким чином зловмисникам вдається

демонструвати операторам «зелені» та безпечні показники під час реальної аварії?

4. У чому полягає асиметрія часу у кіберфізичних системах і чому людина-оператор не завжди здатна врятувати ситуацію?
5. Чому традиційний принцип абсолютної ізоляції мереж не гарантує 100% захисту об'єкта критичної інфраструктури? Наведіть основні вектори його подолання.
6. Яким чином була реалізована первинна доставка шкідливого коду у всесвітньо відомому інциденті зі Stuxnet на ядерний об'єкт в Ірані?
7. Які ключові відмінності між масовими вірусами та таргетованими атаками на КФС?
8. Яку роль відіграють цифрові двійники у сучасній інженерії КФС і як їх можна використовувати для попередження кібератак?
9. Чи можна сьогодні працювати без цифрового двійника?
10. Розкажіть про інцидент із системою TRITON. Чому ця атака вважається новою епохою кіберзагроз і чим вона відрізнялася від попередніх кейсів?
11. Які методи використовують зловмисники, щоб максимально ускладнити відновлення обладнання після диверсії?

Тема 3. СПЕЦИФІКА І ПРОБЛЕМИ ОРГАНІЗАЦІЇ КІБЕРЗАХИСТУ КІБЕРФІЗИЧНИХ СИСТЕМ РЕАЛЬНОГО ЧАСУ

План

1. Вступ
2. Організаційно-технічні парадокси КФС. Організаційна специфіка
3. Часові обмеження та диктатура реального часу

4. Ресурсні асиметрії
5. Шляхи еволюції: від суцільного захисту до легковагової селективності

1. Вступ

Стрімка еволюція кіберфізичних систем, включно з автономними суднами (MASS) та безпілотними літальними апаратами, супроводжується принциповою трансформацією архітектурних ризиків. Якщо традиційні інформаційні технології пріоритезують конфіденційність та цілісність даних, то критичною вимогою для операційних технологій та КФС є безумовна безперервність фізичного процесу в режимі реального часу.

Впровадження систем кіберзахисту в контури управління КФС наштовхується на фундаментальне протиріччя між рівнем безпеки та обмеженнями системи. При цьому специфіка проблеми кардинально залежить від класу та архітектури самої системи - чи маємо ми справу з мобільними бортовими платформами, чи з великомасштабними стаціонарними промисловими об'єктами.

Ключовими аспектами проблеми є:

1. *Функціонування в режимі реального часу:* у контурах керування КФС кожен цикл обробки даних підпорядкований часовому бюджету. Затримка виконання операцій, зумовлена накладними витратами на захисні процедури, призводить до порушення стабільності об'єкта або втрати керованості;
2. *Дуалізм обмежень (залежно від класу КФС):*
 - Для мобільних і периферійних КФС (БПЛА, MASS): ресурсні обмеження (жорсткі ліміти енергоспоживання, маси, габаритів та обчислювальних потужностей), що унеможливорює використання важких ІТ-рішень.
 - Для великомасштабних стаціонарних КФС (SCADA, Smart Grid): архітектурно-життєві обмеження (проблема гетерогенності, застарілого обладнання та обмежень пропускну здатності промислових мереж).
3. *Організаційно-технологічний розрив:*
 - Для мобільних і периферійних КФС (БПЛА, MASS): працюють автономно в умовах віддаленості від баз обслуговування. «Вікна обслуговування» відсутні через фізичну ізоляцію під час місії. Дрон у повітрі або автономне судно посеред океану неможливо «зупинити біля бордюру», вимкнути контролер і залити оновлення. Вони функціонують у режимі самотійного виживання.
 - Для великомасштабних стаціонарних КФС (SCADA, енергомережі): працюють у складі величезних розподілених мереж із тисячами точок підключення, де кожна точка - потенційний вхід для злоумисника. «Вікна обслуговування» відсутні через безперервний комерційний та технологічний цикл (24/7/365). Зупинка лінії нафтопроводу чи турбіни електростанції заради встановлення патча безпеки коштує мільйони і загрожує техногенною аварією.

Метою є системний аналіз специфіки організації захисту в контурах реального часу з урахуванням класу кіберфізичних систем, виявлення джерел накладних затримок та розгляд концептуальних шляхів переходу від суцільного до легковагового і селективного захисту КФС.

2. Організаційно-технічні парадокси КФС. Організаційна специфіка

У традиційних інформаційних системах існує стандартизована практика управління вразливістю: у період мінімального навантаження (наприклад, уночі чи на вихідних) інженери зупиняють сервіси, тестують оновлення, встановлюють патчі безпеки та перезавантажують сервери. Для кіберфізичних систем концепція планового відключення є неприйнятною, проте причини цього різняться залежно від класу системи:

- Для мобільних і периферійних КФС (БПЛА, MASS) зупинка неможлива через фізичну ізоляцію під час місії (дрон у польоті чи судно у відкритому морі).
- Для великомасштабних стаціонарних КФС (SCADA, енергомережі) зупинка неприпустима через безперервний режим роботи 24/7/365 та високу вартість простою технологічного процесу.

Перезапуск керуючого контролера, зупинка посеред виконання маневру чи технологічного циклу рівнозначна втраті керованості.

Виявлення критичної вразливості у програмному забезпеченні КФС створює нерозв'язну дилему: залишити систему відкритою для атаки чи спробувати оновити її на ходу, ризикуючи спровокувати аварійну зупинку процесу через збій у роботі «живого» контуру.

Це одне з найболючіших питань практичної інженерії безпеки в ОТ. Цей парадокс розв'язується за допомогою чотирьох ключових стратегій:

1. Архітектурне резервування.

Якщо система критична і не може зупинитися ні на секунду, апаратна база будується за принципом надлишковості. Це працює наступним чином. У нас є два абсолютно ідентичних контролери: основний і резервний. Коли потрібно встановити патч на основний контролер, система плавно перемикає все навантаження та управління на резервний. Основний контролер безпечно відключається від процесу, оновлюється, проходить перевірку і повертається в режим резерву або стає головним. Усі маніпуляції відбуваються без зупинки самого фізичного процесу.

2. Компенсаційні заходи замість патчу.

Дуже часто для застарілого облановлення фізично неможливо випустити патч, оскільки виробник заклав компанію або застаріла мікроархітектура не підтримує зміни. У такому разі вразливість «закривають ззовні»: ми не чіпаємо сам контролер, але ставимо перед ним інтелектуальний апаратний шлюз (промисловий фаєрвол або систему виявлення вторгнень), який на рівні мережі блокує будь-які спроби експлуатації цієї вразливості. Програма в контролері залишається з «діркою», але зловмисники фізично не можуть до неї дістатися через мережеві фільтри.

3. Попереднє тестування на цифрових двійниках та стендах.

Ризик аварії під час оновлення часто виникає через те, що патч не був повноцінно протестований в умовах, наближених до реальних. Тому перш ніж залити оновлення на реальний об'єкт (наприклад, на борт MASS), його проганяють через повну віртуальну модель. Цифровий двійник моделює мільйони сценаріїв поведінки системи з новим кодом, і лише після 100% успішного проходження симуляції оновлення дозволяють до встановлення.

4. Hot-Patching (Гаряче пропагування пам'яті)

Це складний технологічний метод точкового втручання, який використовується в сучасних операційних системах реального часу. Замість повної заміни прошивки і перезавантаження контролера, система пропагує лише невелику ділянку машинного коду безпосередньо в оперативній пам'яті льоту, перенаправляючи виконання на виправлену функцію. Це вимагає ювелірної точності, але дозволяє закрити критичну вразливість за мікросекунди без зупинки процесу.

Саме поєднання цих підходів дозволяє підтримувати КФС, не наражаючи фізичний об'єкт на небезпеку.

Завдання. Як ви вважаєте, який із цих методів є найскладнішим для впровадження на борту безпілотних систем (БПЛА чи MASS)?

Архітектура КФС часто формується за принципом *гетерогенності*, де поєднуються компоненти різного віку та призначення. Проте прояв цієї проблеми суттєво відрізняється за класами систем:

- У стаціонарних промислових КФС (SCADA, Smart Grid) ключовим є застаріле залізо із проєктним терміном служби 15–25 років, яке використовує незашифровані протоколи і фізично не має ресурсів для сучасного захисту. Повна модернізація такого парку «на льоту» є економічно нездійсненною.
- У мобільних КФС (БПЛА, MASS) проблема гетерогенності частіше виникає через швидку еволюцію бортових модулів: новітні мікрокомп'ютери зі штучним інтелектом сусідять зі старішими датчиками та контурами стабілізації, створюючи конфлікти інтерфейсів та обчислювальних можливостей.

Проблема гетерогенності та «старого заліза» залишається гострою. Це відбувається завдяки тому, що на практиці повне швидке оновлення вкрай важко реалізувати з кількох причин:

- ніхто не зупиняє роботу цілого підприємства чи флоту, щоб одночасно викинути все старе обладнання і поставити нове. Заміну роблять еволюційно: замінили один зношений контролер або додали новий модуль навігації, а сусідні блоки залишили старими, бо вони ще працюють. У результаті замість гарної єдиної системи виходить «лоскутна ковдра» (гетерогенність);
- промислові контролери, бортові обчислювачі суден чи системи керування енергомережами коштують шалених грошей. Бізнес прагне вичавити з купленого обладнання максимум до останнього, доки воно виконує свою базову фізичну функцію;

- у критичних галузях (морський транспорт, авіація, енергетика) заміна будь-якого елемента керування це не просто купівля нової плати в магазині. Це місяці або й роки проектування, узгоджень із класифікаційними товариствами, стендових випробувань та сертифікації безпеки.

Тобто залізо дійсно можна оновити швидше, ніж за 25 років, але сам процес оновлення відбувається хвилями й фрагментарно. Поки оновлюється одна частина системи, інша старіє. І цей безперервний цикл несинхронних змін якраз і породжує ту саму архітектурну гетерогенність та вразливості.

3. Часові обмеження та диктатура реального часу

У кіберфізичних системах управління фізичним об'єктом підпорядковане законам не цифрового, а матеріального світу. Це зумовлює перехід від абстрактного часу комп'ютерного обчислення до поняття жорсткого реального часу, де запізнення виконання команди може бути рівнозначним її невиконанню або катастрофі.

3.1. Поняття часового бюджету

Кожен замкнений цикл управління в КФС (наприклад, цикл стабілізації польоту безпілотної або утримання курсу MASS) працює за незмінним алгоритмом:



Весь цей ланцюжок має суворо обмежений час виконання - часовий бюджет, який зазвичай вимірюється мілісекундами або навіть мікросекундами. Якщо загальний час циклу становить, наприклад, 10 мс, то система зобов'язана завершити всі обчислення рівно за цей час.

3.2. Накладні витрати на захист

Будь-який механізм кібербезпеки (шифрування трафіку, цифрові підписи пакетів, глибокий аналіз пакетів (DPI)) вимагає додаткових обчислювальних ресурсів і, відповідно, часу. У корпоративному ІТ додаткові 5-10 мілісекунд на перевірку безпеки залишаються непомітними. У КФС ці додаткові мілісекунди віднімаються безпосередньо з часового бюджету керуючого циклу. Якщо процедури захисту починають «з'їдати» критичну частину часу, контролер перестає встигати обробляти актуальні дані від датчиків.

3.3. Катастрофічні наслідки затримки

Коли захист створює надмірну затримку, система стикається з ефектом втрати детермінізму: контролер приймає рішення на основі застарілої інформації про положення об'єкта в просторі, бо обчислення затрималися через, наприклад, крипто перетворення; у динамічних об'єктах (наприклад, швидкісних БПЛА) затримка реакції призводить до накопичення похибки регулювання, розгойдування системи та її руйнування або аварії.

Таким чином, у КФС захист не може бути «будь-якою ціною». *Будь-який алгоритм безпеки підлягає жорсткому лімітуванню у КФС: його часова складність повинна апріорі вписуватися в найгірший сценарій часового бюджету системи.*

На перший погляд здається, що якщо криптографії чи аналізу трафіку потрібно більше часу, можна було б «пом'якшити» вимоги до реального часу функціонування КФС (наприклад, замість циклу в 10 мілісекунд зробити цикл у 100 мілісекунд). Однак, у кіберфізичних системах розширити часовий ліміт неможливо, тому що його диктує не розробник софту і не процесор, а сама фізика керованого об'єкта. «Уповільнити» систему заради безпеки неможливо, оскільки:

- Часовий бюджет контуру управління залежить від того, наскільки швидко змінюється фізичний стан об'єкта. Наприклад, швидкісний безпілотник або ракета рухаються в повітрі з величезною динамікою. Вітер змінює кут атаки за мілісекунди. Якщо датчики й контролер реагуватимуть повільно (з розширеним часовим бюджетом), апарат просто втратить стійкість і впаде ще до того, як система встигне порахувати нову коригувальну команду.
- У теорії автоматичного керування є залізне правило: частота опитування датчиків і керуючих дій має бути в кілька разів вищою за найвищу частоту процесів, які ми намагаємося контролювати (Теорема Найквіста-Котельникова). Якщо ми збільшуємо часовий ліміт циклу, ми знижуємо частоту дискретизації. Система стає «сліпою» до коротких, але критичних імпульсів і збурень у середовищі (наприклад, різкий удар хвилі в борт судна або раптовий порив вітру). Між двома тактами такого «повільного» контролера може статися аварія, яку система навіть не помітить вчасно.
- Кіберфізична система відрізняється від ІТ тим, що вона взаємодіє з реальним матеріальним світом. Процесор у комп'ютері може «задуматися» на долю секунди, і файл просто завантажиться пізніше. Але падіння безпілотника, зіткнення судна чи вибух котла під тиском підпорядковуються законам механіки та термодинаміки, яким абсолютно байдуже, що алгоритм безпеки виконує складне 256-бітне шифрування.

Таким чином, *реальний фізичний світ неможливо змусити «йти повільніше» чи «почекати», поки відпрацює захист. Саме тому неможливо просто розширити часові бюджети, а треба вирішувати зворотну, набагато складнішу задачу: підганяти алгоритми захисту під*

суворі ліміти реального часу, використовуючи легковагові та селективні методи.

4. Ресурсні асиметрії

У класичному ІТ захист інформації спирається на потужні сервери, необмежені обсяги оперативної пам'яті та стабільне живлення від стаціонарної електромережі. У світі кіберфізичних систем забезпечення безпеки стикається з глибокою апаратною та архітектурною асиметрією, характер якої безпосередньо залежить від класу системи.

1) Мобільні та периферійні КФС: SWaP-асиметрія (Size, Weight, and Power)

Для рухомих об'єктів (БПЛА, MASS, робототехніки) головним лімітуючим фактором є концепція SWaP:

- Живлення здійснюється від бортових акумуляторів. Будь-який надлишковий обчислювальний процес (наприклад, складне шифрування чи важкі протоколи аутентифікації) навантажує процесор, збільшує енергоспоживання і скорочує час автономної місії (наприклад, зменшує час польоту дрона з 40 до 30 хвилин);
- У герметичних бортових корпусах відсутні активні системи охолодження. Потужні обчислення викликають перегрів чипів і переходять в аварійний режим скидання частот;
- Бортові мікроконтролери розраховані на виконання чітких циклічних математичних алгоритмів керування, а не на трудомісткі захисні стеки.

2) Великомасштабні стаціонарні КФС: Архітектурна Legacy-асиметрія

Для стаціонарних промислових об'єктів (Smart Grid, енергомережі, підприємства) проблеми енергії чи ваги немає, але з'являється інший бар'єр - застаріле залізо та гетерогенність:

- Контролери на технологічних об'єктах встановлюються на десятиліття. Багато з них фізично не мають ані криптографічних співпроцесорів, ані пам'яті для впровадження сучасних стандартів безпеки;
- Промислові канали зв'язку часто є вузькими і повільними, тому передача додаткових пакетів безпеки чи сертифікатів створює критичні затори в мережі.

3) Нереалістичність масштабування

У хмарному чи корпоративному ІТ при зростанні навантаження адміністратор просто додає новий сервер чи збільшує обсяг оперативної пам'яті. У КФС горизонтальне чи вертикальне масштабування заблоковане:

- На борту БПЛА неможливо «доставити» ще два процесори, бо це порушить балансування та масу;
- У стаціонарній системі заміна тисяч розподілених датчиків вимагає колосальних капітальних витрат і зупинки технологічного процесу.

Таким чином, проектування захисту в КФС не допускає підходу «грубої сили». Для мобільних систем захист має бути екстремально легковаговим

заради збереження заряду та маси, а для стаціонарних - адаптованим під роботу з наявним архаїчним залізом через зовнішні компенсаційні засоби.

4) Безпеки в реальному світі КФС

Приклад А. Тактичний БПЛА. Невеликий розвідувальний дрон, що працює на літій-полімерному акумуляторі з часом автономного польоту 40 хвилин. Якщо інженери спробують накласти на кожен пакет передачі телеметрії важкий протокол шифрування комерційного рівня (наприклад, багатоступеневий TLS з важким обміном ключами на слабкому бортовому процесорі), процесор буде постійно завантажений на 100%. Додаткове навантаження на процесор і радіомодуль збільшує енергоспоживання. Час польоту дрона скорочується з 40 хвилин до 30. Батарея сідає раніше, ніж дрон виконує місію. У цьому випадку надлишковий захист призводить до фізичної втрати об'єкта.

Приклад Б. MASS у відкритому морі. На корпусі судна встановлені десятки датчиків вітру, тиску і положення руля, підключені через шину до локального мікроконтролера. Ці датчики працюють на енергозберігаючих протоколах і мають мікроскопічні обчислювальні потужності. Повна модернізація кожної точки збору даних під стандарти сучасної криптографії неможлива, оскільки чипи датчиків просто не мають вбудованих криптографічних співпроцесорів.

5. Шляхи еволюції: від суцільного захисту до легковагової селективності

5.1. Крах парадигми «Захистимо все і суцільно»

У класичному корпоративному ІТ панує доктрина інформаційної безпеки - ешелонована оборона та принцип «перевірйй усе, шифруй усе». Там прийнято шифрувати весь трафік (наприклад, масове використання TLS 1.3), пропускати кожен пакет через системи глибокого аналізу трафіку (DPI) та вимагати багатофакторної аутентифікації для будь-якої дії.

У кіберфізичних системах спроба перенести цю парадигму в чистому вигляді призводить до парадоксального ефекту: *заходи безпеки при такому використанні можуть знищити саму систему ефективніше за будь-яку кібератаку*. Чому так відбувається? Промислові та бортові протоколи зв'язку розраховані на передачу крихітних пакетів даних (наприклад, 8–64 байти в шині CAN або Modbus) із мінімальною затримкою. Якщо обгорнути кожен такий мікропакет у важкі заголовки безпеки, обсяг службового трафіку в рази перевищить обсяг корисних даних. Мережа просто захлиснеться від власних «добитків безпеки». Крім цього, канали зв'язку на борту рухомих об'єктів або в розподілених промислових мережах мають суворо лімітовану пропускну здатність. Забивання їх суцільним шифруванням та перевірками залишає критичні команди керування без частотного ресурсного забезпечення.

Приклад А. Тактичний безпілоти́к (Мобільний клас КФС). Припустимо, що концепція «захистити все і суцільно» застосовується до внутрішньої мережі датчиків та контролерів швидкісного БПЛА. Бортовий процесор налаштовують на обов'язкове криптографічне підписання та

шифрування кожного пакета датчика висоти, який оновлюється з частотою 1000 разів на секунду. Малопотужний бортовий мікроконтролер витрачає 80% свого часу не на розрахунок траєкторії польоту, а на виконання криптографічних операцій. Через це затримується обробка чергового пакета від інерціальної навігаційної системи. Літак втрачає миттєву стабілізацію при пориві вітру і йде в некероване пікірування. Захист «вбив» об'єкт.

Приклад Б. Розподілена енергопідстанція (Стаціонарний клас КФС)

Промислова мережа на підстанції Smart Grid, яка передає команди вимикачами високої напруги за протоколом IEC 61850. На всьому масиві мережі встановлюють систему глибокого аналізу трафіку (DPI), яка намагається розібрати й перевірити кожен пакет на наявність аномалій. Через брак обчислювальної потужності на проміжних шлюзах виникає черга пакетів. Команда аварійного відключення лінії при короткому замиканні затримується всього на 20-30 мілісекунд через те, що фаєрвол «думав» над перевіркою безпеки. За ці мілісекунди обладнання підстанції зазнає фізичного руйнування від електричної дуги.

Таким чином, ***безпека в КФС не може бути рівномірною і тотальною.*** Спроба перенести за лекалами корпоративного ІТ принцип «шифруй усе, перевіряй усе» призводить до того, що ***сам засіб захисту стає головною загрозою для фізичного виживання КФС.*** Накладні витрати на обробку, мережеві затори та втрата детермінізму через затримки можуть «вбити» керованість швидше за будь-який варіант кібератаки.

5.2. Концепція легковагової криптографії (Lightweight Cryptography).

Якщо традиційні криптографічні алгоритми (такі як класичний RSA або важкі режими AES) вимагають занадто багато пам'яті, процесорного часу та енергії, це не означає, що КФС мають залишатися повністю «голими» перед загрозою перехоплення чи підробки команд. Виходом із цієї дилеми стала ***легковагова криптографія (Lightweight Cryptography, LWC).***

Легковагова криптографія - це клас криптографічних примітивів (блочних шифрів, хеш-функцій, алгоритмів автентифікації), спеціально розроблених для роботи в умовах екстремальних ресурсних обмежень (периферійні пристрої, IoT, бортові контролери БПЛА та датчики тощо).

Головна мета LWC - забезпечити достатній рівень захисту інформації за мінімальних витрат обчислювальної потужності, оперативної пам'яті та енергії.

Ключові відмінності LWC від класичної криптографії

- ***Мінімальний розмір коду та пам'яті:*** алгоритми LWC оптимізовані так, щоб уміщатися в крихітні обсяги пам'яті мікроконтролерів;
- ***Енергоефективність:*** кількість енергії, витрачена на шифрування одного байта даних, знижується в рази, що критично для пристроїв на батарейках (наприклад, мобільних КФС).
- ***Адаптація під апаратну або програмну реалізацію:*** багато алгоритмів LWC проєктуються з урахуванням можливості їх надзвичайно

ефективної реалізації як на програмному рівні, так і на рівні крихитних схем FPGA/ASIC.

Довгий час існувало безліч експериментальних легковагових шифрів, проте у 2023 році Національний інститут стандартів і технологій США (NIST) офіційно завершив багаторічний конкурс і обрав переможця для стандартизації у сфері LWC - алгоритм Ascon. Ascon використовує так звану структуру губки і забезпечує як шифрування, так і автентифікацію даних. Він створений спеціально для того, щоб захищати легкі пристрої (наприклад, датчики в промислових мережах чи бортові контролери), які не здатні потягнути стандартний важкий AES.

Легковагова криптографія потрібна для обох класів КФС, але причини її застосування та «болючі точки» дещо відрізняються залежно від архітектури системи:

- **Для мобільних і периферійних КФС**

Бортові контролери цих систем працюють від акумуляторів і мають мінімальну обчислювальну потужність. Застосування класичної криптографії (наприклад, важких протоколів TLS або складних блочних шифрів) швидко розряджає батарею або перегріває процесор. Легковагові алгоритми (як-от *Ascon*) дозволяють захистити канали зв'язку та команди управління з мінімальними витратами енергії та пам'яті.

- **Для великомасштабних стаціонарних КФС**

Хоча в диспетчерських пунктах таких систем стоять потужні промислові сервери, на «низовому» рівні (на самих датчиках, засувках і кінцевих контролерах) працюють прості, можливо застарілі мікроконтролери. Вони фізично не мають ресурсів для важких алгоритмів безпеки. Крім того, промислові мережі часто мають обмежену пропускну здатність. LWC дозволяє захистити ці вразливі периферійні точки без необхідності повністю міняти все масивне залізо підприємства.

Таким чином, легковагова криптографія є універсальним інструментом для обох класів КФС, оскільки в обох випадках ключовою проблемою є дефіцит ресурсів на периферії (будь то через живлення від батареї на дроні, чи через старе залізо на промисловій підстанції).

5.3. (Легковагова) стеганографія для КФС

У традиційному ІТ стеганографія часто асоціюється з приховуванням картинки в картинці чи важких файлів. У кіберфізичних системах усе дещо інакше. Критичні службові мітки, контрольні суми або елементи автентифікації ховаються всередину вже існуючих пакетів телеметрії (наприклад, у молодші біди координат GPS, потокового відео з бортової камери тощо). Зашифрований трафік у мережі завжди кидається в очі супротивнику. Стеганографія дозволяє замаскувати керуючі або захисні дані під звичайний рутинний шум датчиків, що ускладнює перехоплення та активне придушення каналу зв'язку.

Переваги для мобільних КФС:

- На відміну від криптографії, яка додає до пакетів заголовки, теги автентифікації, стеганографія використовує наявні поля даних. Це критично для мобільних КФС (БПЛА, MASS), де кожен додатковий байт службового трафіку в умовах радіоелектронної боротьби (РЕБ) чи вузьких каналів є критичним;
- Коли порушник намагається виявити канали управління за наявністю криптографічних сигнатур, стеганографічне приховування команд усередині відкритих потоків телеметрії або відеопотоку знижує ймовірність виявлення та блокування каналу зв'язку.

Якщо для мобільних систем (БПЛА, MASS) застосовується саме легковагова стеганографія, яка передусім є засобом маскування в радіоефірі та економії ресурсів каналу, то для великомасштабних стаціонарних КФС роль стеганографії дещо інша. У стаціонарних системах немає проблеми радіоелектронної боротьби, але є свої критичні точки, про які вже згадувалося в попередніх темах: застарілі протоколи, жорсткі вимоги до сумісності та загроза прихованих атак. Але стеганографія знаходить своє застосування і у стаціонарних КФС (для стаціонарних КФС вимога «легковаговості» до стеганографії чи обробки даних не є ключовою):

1) *Неінвазивне маркування та передача метаданих у старих протоколах.* Старі промислові протоколи (наприклад, Modbus) розроблялися десятиліттями тому і не мають вбудованих полів для безпеки, міток цілісності чи сервісної аутентифікації. Якщо додати нові заголовки, старі контролери просто «злякаються» і відкинуть пакет як помилковий. Використання молодших бітів у регулярних пакетах телеметрії для прихованого (стеганографічного) передавання невеликих міток цілісності чи діагностичних статусів дозволяє покращити контроль без зміни стандарту протоколу, не ламаючи застаріле залізо.

2) *Стійкі резервні канали зв'язку.* У великих розподілених мережах (наприклад, енергомережах) магістральні канали зв'язку можуть піддаватися цілеспрямованим атакам або блокуванню. На випадок компрометації основних каналів керування, легковагові вбудовані канали (коли критичні службові флаги чи аварійні команди «розчиняються» у фоновому масиві звичайних, нічим не примітних технологічних вимірювань) можуть використовуватися як прихований резервний канал зв'язку між підстанціями та диспетчерським центром.

3) *Цифрові водяні знаки для цілісності логів і телеметрії.* На стаціонарних об'єктах збираються терабайти логів та відеопотоків із камер спостереження за периметром. Зловмисник може спробувати непомітно підробити історичні дані датчиків, щоб приховати наслідки аварії чи маніпуляцій. Вбудовування легковагових криптографічних водяних знаків безпосередньо в структуру поточкових даних та архівів дозволяє миттєво виявити факт несанкціонованої модифікації інформації в базах даних або журналах подій.

4) *Виявлення прихованих каналів зловмисника.* У стаціонарних КФС фахівці з безпеки частіше стикаються не з тим, щоб самим використовувати

стеганографію, а з тим, щоб боротися з нею. Зловмисники часто використовують стеганографію в промисловому трафіку (наприклад, ховають команди управління або виводять дані через нестандартні відхилення в патернах передачі Modbus), щоб обійти промислові фаєрволи. Тому на стаціонарних об'єктах критично важливим є стеганоаналіз промислових мереж.

5.4. Селективні методи захисту. Основний принцип - захищати не весь потік даних суцільно, а лише критичні інваріанти та пріоритетні команди управління. Захист концентрується виключно на критично важливих елементах, адаптуючись до специфіки класу системи.

1) Стратегія селективності для мобільних і периферійних КФС. Захищаються критичні команди виконавчих механізмів (наприклад, аварійне повернення, зміна курсу, підтвердження місії) та життєво важливі канали зв'язку. Для захисту використовується легковагова криптографія і легковагова стеганографія для маскування службових міток у потоках телеметрії. При цьому жертвуємо рутинною телеметрією (наприклад, вторинні показники температури двигуна чи незначні похибки сенсорів), що передається у відкритому вигляді або з мінімальною цілісністю. Її короточасне перехоплення чи спотворення не веде до фізичної катастрофи, зате економить заряд батареї та обчислювальний час процесора (зберігаючи часовий бюджет у мілісекундах).

2) Стратегія селективності для стаціонарних промислових КФС. Захищаються ключові магістральні вузли передачі даних, центри диспетчеризації та вхідні шлюзи на рівнях керування високою напругою чи потоками речовин. Для захисту використовуються зовнішні компенсаційні засоби (промислові фаєрволи, системи глибокого аналізу трафіку на мережних рівнях, які не чіпають слабкі контролери напряму) та архітектурне резервування. При цьому рівень локальних датчиків і застарілих контролерів залишається архаїчним усередині локального сегмента, оскільки їх фізично неможливо оновити. Безпека тут забезпечується «периметром навиворіт» і мережевою сегментацією, що захищає їх ззовні без втручання в застарілий код.

Таким чином, безпека кіберфізичних систем реального часу - це завжди балансування на межі компромісів. Фахівець з кіберзахисту в КФС не має права сліпо копіювати методи корпоративного ІТ. Зважаючи на клас системи (будь то маневрений дрон у небі чи масивна енергомережна підстанція на землі), фахівець повинен чітко розраховувати часові бюджети, враховувати апаратні обмеження, уникати пастки суцільного захисту та застосовувати легковагові й селективні методи там, де кожна мілісекунда затримки означає фізичну аварію.

Питання

1. На які два базові класи поділяються кіберфізичні системи? Чому цей поділ є фундаментальним для організації кіберзахисту?

2. У чому полягає організаційний «парадокс патчингу» у КФС порівняно зі стандартними процедурами оновлення в класичному ІТ?
3. Чим принципово відрізняються причини відсутності «вікон обслуговування» для мобільних КФС та стаціонарних промислових?
4. Які чотири стратегії пропонуються для усунення дилеми між необхідністю безпеки та неприпустимістю зупинки «живого» контуру КФС?
5. Як проблема гетерогенності та застарілого обладнання впливає на безпеку стаціонарних промислових систем?
6. Що таке часовий бюджет у контурах керування КФС і з яких етапів складається один цикл обробки даних?
7. Чому накладні витрати на безпеку, які є непомітними у корпоративному ІТ, можуть призвести до аварії чи втрати керованості в КФС?
8. Яким чином теорія Найквіста-Котельникова обмежує можливість довільного збільшення часу на виконання процедур захисту?
9. Які три ключові обмеження входять до концепції SWaP-асиметрії, і чому вони критичні для БПЛА та MASS?
10. Чому в кіберфізичних системах неможливо застосувати звичний для хмарного ІТ підхід горизонтального чи вертикального масштабування для розширення потужностей безпеки?
11. Чому спроба перенести парадигму «захистити все і суцільно» руйнує продуктивність і детермінізм КФС?
12. До яких наслідків для фізичного об'єкта може призвести ефект вичерпання пропускну здатності у промислових мережах?
13. Які технічні переваги роблять легковагову криптографію (зокрема стандарт Ascon) ключовим інструментом захисту периферійних пристроїв КФС порівняно з класичним AES чи RSA?
14. У чому полягає специфіка застосування саме легковагової стеганографії для мобільних КФС (БПЛА, MASS) в умовах радіоелектронної протидії?
15. Що є головною суттю концепції селективного захисту КФС?

Тема 4. КІБЕРФІЗИЧНІ СИСТЕМИ: АДАПТАЦІЯ ІСНУЮЧИХ СИСТЕМ ЗАХИСТУ ТА НОВІ КРИТЕРІЇ СТІЙКОСТІ НА ПРИКЛАДІ МОРСЬКИХ АВТОНОМНИХ ОБ'ЄКТІВ

План

1. Трансформація морської галузі та недієздатність класичних ІТ-шаблонів у кіберфізичних контурах MASS.
2. Стеганографія як ключовий елемент кібербезпеки MASS.
3. Класичні критерії стійкості стеганосистеми та обґрунтування їх ієрархії для MASS.
4. Нові специфічні критерії стійкості для MASS та їхній антагонізм.

1. Трансформація морської галузі та недієздатність класичних ІТ-шаблонів у кіберфізичних контурах MASS. Судно як кіберфізична система.

Історично морська галузь вирізнялася високим ступенем консерватизму, де пріоритет традиційно надавався засобам суто фізичної та навігаційної безпеки, тоді як питання інформаційного захисту сприймалися менеджментом як другорядні, «офісні» завдання, відірвані від реальних процесів у морі. Наслідком такої системної недооцінки став каскад руйнівних кібератак на критичну інфраструктуру світового флоту та портові комплекси, що завдало збитків на сотні мільйонів доларів і довело вразливість сучасних суден.

З появою морських автономних надводних суден (MASS) ситуація набула якісно нового, гіпертрофованого значення. Сучасні фахівці з кібербезпеки часто працюють у відриві від специфіки морської індустрії, намагаючись механічно переносити шаблони корпоративного ІТ-захисту. Такі шаблони стають абсолютно недієздатними в умовах обмеженого часу, нестабільності супутникових каналів зв'язку та повної ізоляваності судна в океані.

Перехід до вищих рівнів автономності перетворив судно на високоінтегровану кіберфізичну систему, яка критично залежить від безперервних потоків вхідних сенсорних даних і зовнішніх команд керування. Повна відсутність або мінімальність екіпажу на борту позбавляє систему «останнього рубежу» ручного захисту. Будь-яке несанкціоноване інформаційне втручання (атаки на канали зв'язку, підміна координат чи модифікація телеметрії) миттєво стає тригером для каскадного руйнування фізичної безпеки судна.

Таким чином, у контурі MASS інформаційна та кібербезпека є первинним, базисним елементом загальної живучості. Фізична безпека автономного судна тепер є лише похідною від його інформаційної безпеки: *судно пливе, якщо дані захищені; якщо дані скомпрометовані - судно гине.*

Актуальність проблеми забезпечення кібербезпеки посилюється набранням чинності Міжнародного кодексу безпеки для MASS (MASS Code від IMO), який вперше на глобальному рівні закріпив вимогу того, що

цифрова зв'язність, надійність програмного забезпечення та кібербезпека є невіддільними елементами живучості судна.

Проте існуючі базові документи (такі як Резолюція IMO MSC.428(98) щодо управління кіберризиками або циркуляр MSC-FAL.1/Circ.3/Rev.2) створювалися переважно для звичайних суден з екіпажем. Вони мають декларативний рекомендаційний характер і не враховують специфіки автономного ШІ-керування в реальному часі.

Оскільки MASS Code базується на цільовому підході і не регламентує конкретних інженерних рішень, у галузі виник гострий методологічний вакуум: існуючі стандарти захисту повністю ігнорують вплив малих інформаційних збурень на реальну фізичну траєкторію судна.

2. Стеганографія як ключовий елемент кібербезпеки MASS

Серед усіх напрямків кібербезпеки автономного судноплавства особливе місце посідає стеганографія, що зумовлено унікальною специфікою автономного судна як об'єкта захисту. На відміну від судна з екіпажем, де втрата або компрометація керуючої команди може бути компенсована рішенням екіпажу на місці, автономне судно часто позбавлене цієї можливості, особливо у разі повної відсутності екіпажу – управляюча команда з берегового центру управління є єдиним джерелом рішень у позаштатній ситуації.

Це породжує на практиці принципово нову вимогу до захисту каналу управління: необхідно приховати сам факт комунікації, оскільки противник, який здійснює моніторинг каналу зв'язку, при виявленні управляючого трафіку отримує можливість його знищення або підміни з прямими наслідками для безпеки судна, вантажу і навколишнього середовища.

Необхідно відзначити, що приховані канали в морському середовищі хоча вже є реальністю, але поки що переважно як інструмент противника: дослідження демонструють реалізованість прихованих каналів на основі Automatic Identification System (AIS) для організації атак на морську інфраструктуру. Це лише підкреслює актуальність симетричного застосування стеганографічних методів на користь захисту каналу управління автономним судном.

Стеганографічний канал, що забезпечує приховування факту передачі керуючої інформації, є найбільш адекватною відповіддю на цю специфічну загрозу, що визначає актуальність стеганографії як самостійного напрямку кібербезпеки КФС, зокрема автономного судноплавства.

Властивості прихованого (стеганографічного) каналу зв'язку, що організується, визначаються властивостями стеганографічної системи, що лежить в його основі. Однією з основних вимог до стеганосистеми є вимога стійкості, яка має тут безліч складових. Однак сукупність всіх цих складових не є вичерпною, а отже, не гарантовано достатньою для забезпечення стійкості стеганосистеми в MASS. Існуючі критерії стійкості стеганосистеми сформовані без урахування специфіки її використання у складі кіберфізичної системи реального часу. Жорстка часова регламентація інформаційних

потоків у системах управління автономним судном, а також можливий з великою ймовірністю автоматизований характер генерації контейнера для прихованої передачі управляючої інформації відкривають нові вектори виявлення стеганографічного каналу, не передбачені існуючими критеріями стійкості. Стеганосистема, спроектована без урахування цих векторів, залишається вразливою незалежно від ступеня забезпечення існуючих критеріїв, що в умовах автономного судноплавства безпосередньо тягне за собою компрометацію каналу управління судна з прямими фізичними наслідками.

Таким чином, стеганографічний канал, що забезпечує повне приховання самого факту передачі керуючої інформації, є найбільш адекватним і технологічно необхідним відповідником на специфічні кіберзагрози в середовищі MASS, оскільки в умовах повної автономії судно не може покладатися на локальні рішення команди на борту, надійність доставки команд із берегового центру (ROC) стає критичним фактором виживання. Стеганографічне приховування перетворює канал зв'язку на захищений «невидимий міст», що мінімізує ризики перехоплення управління. Крім цього, з огляду на те, що деструктивні сили вже активно використовують приховані канали в морському середовищі (наприклад, маніпуляції через системи AIS), впровадження власних стеганографічних систем є не просто бажаним покращенням, а єдиним адекватним симетричним інструментом захисту критичної інфраструктури безпілотного флоту.

3. Класичні критерії стійкості стеганосистеми та обґрунтування їх ієрархії для MASS

Розглянемо детально існуючі критерії стійкості стеганосистеми, структурна схема якої представлена на рис.4.1:

- K_1 - стійкість до виявлення факту існування прихованої інформації: забезпечення непомітності стеганографічного каналу на тлі легітимного трафіку;
- K_2 - стійкість до витягу приховуваної інформації: захист від несанкціонованого видобування прихованих даних стороннім спостерігачем;
- K_3 - стійкість до нав'язування хибних повідомлень по стеганографічному каналу (імітостійкість): захист від впровадження супротивником власних фальсифікованих команд у канал зв'язку;
- K_4 - стійкість до відновлення секретного ключа стеганосистеми: захист базису стеганографічної стійкості за принципом Кірхгоффа;
- K_5 - стійкість до атак проти вбудованого повідомлення: захист від знищення або спотворення корисного навантаження внаслідок завад чи активної протидії.

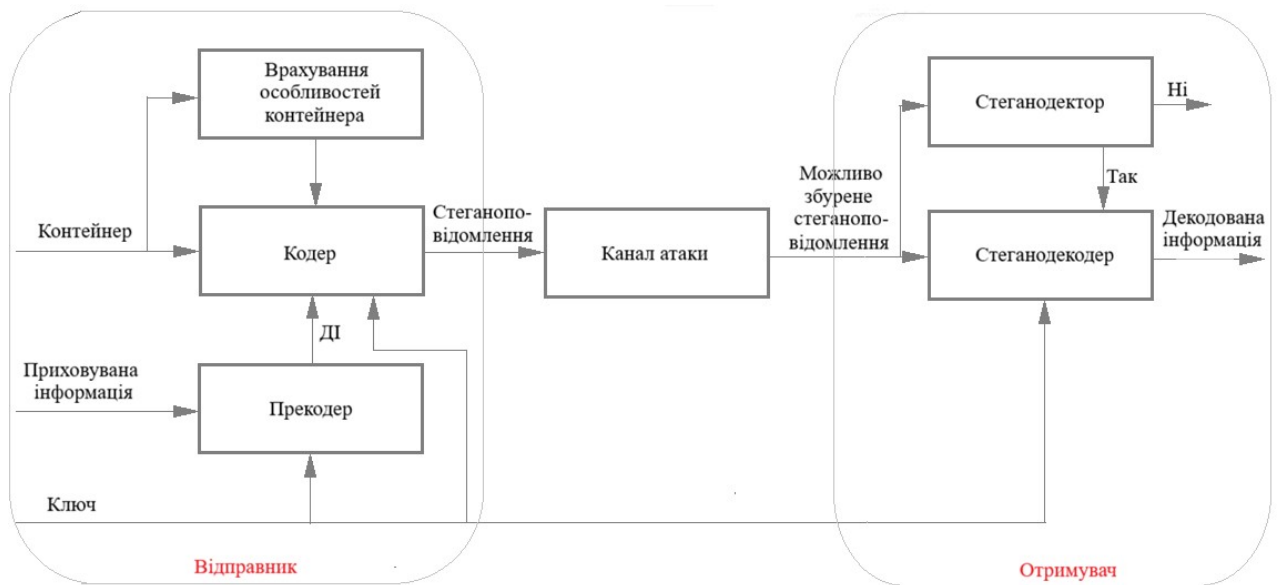


Рис.4.1. Узагальнена структурна схема стеганосистеми

Забезпечення кожного з цих критеріїв необхідне для стеганосистеми, що використовується в структурі MASS. Справді, якщо противник ідентифікує конкретний потік даних як стеганографічний канал управління, він отримує можливість цілеспрямовано атакувати саме його: заблокувати, підмінити або знищити керуючі команди.

K_1 забезпечує невідмінність стеганографічного каналу від легітимного трафіку, позбавляючи противника можливості такої ідентифікації, з чого випливає необхідність критерію K_1 . З урахуванням того, що керуючі команди є критично важливою інформацією, їх розкриття дозволяє противнику отримати повне уявлення про наміри та дії оператора судна, що говорить про необхідність забезпечення критерію K_2 . Автономне судно у разі повної відсутності екіпажу беззастережно виконує отримані команди. Нав'язування хибної керуючої команди безпосередньо тягне за собою виконання судном дій на користь противника. Недопущенню цього є забезпечення критерію K_3 . Компрометація стеганографічного ключа (рис.1) тягне повну компрометацію стеганосистеми - противник знає про цю систему все, може робити як активні, так і зловмисні дії, що говорить про необхідність забезпечення критерію K_4 . Знищення або спотворення команди в умовах автономного судна означає втрату управління, що може мати критичні негативні наслідки для його функціонування і аргументує необхідність забезпечення критерію K_5 .

В наявній множині критеріїв стійкості стеганосистеми $K = \{K_1, K_2, K_3, K_4, K_5\}$ для забезпечення штатного функціонування MASS найбільш критичним є K_3 . Розглянемо це докладно.

Головна методологічна відмінність MASS від звичайної інформаційної системи полягає в тому, що це є кіберфізичний об'єкт. Тут компрометація інформації має значення тоді, коли вона веде до зміни фізичного стану судна. Концепція оцінки ризиків для MASS суттєво відрізняється від традиційних

ІТ-шаблонів. Для кіберфізичного об'єкта, яким є автономне судно, компрометація інформації (зокрема прихованих каналів зв'язку) розглядається як критична загроза виключно тоді, коли вона призводить до деструктивної зміни його фізичного стану (параметрів руху, траєкторії тощо). Якщо інформаційне втручання порушника не викликає аномальних реакцій виконавчих механізмів, така атака не порушує живучість об'єкта. З огляду на це, зіставлення критеріїв стійкості $K = \{K_1, K_2, K_3, K_4, K_5\}$ стеганосистеми у специфічному полі морського автономного судноплавства дозволяє довести безумовний пріоритет критерію імітостійкості K_3 . На відміну від критеріїв K_1 і K_2 , які відповідають пасивному порушнику, саме злам K_3 призводить до активного перехоплення керування об'єктом, неавторизованої зміни його фізичного стану. Компрометація критерія стійкості до відновлення ключа K_4 , який за принципом Кірхгоффа виступає базисом захисту стеганосистеми, визначає потенційну можливість зламів, тоді як руйнування K_3 трансформує цю можливість у незворотну фізичну аварію. Наслідком порушення K_5 є втрата зв'язку (яка взагалі теж може призвести до негативних фізичних наслідків, що розглядається нижче), тоді як порушення K_3 веде до легітимізації системами судна фальсифікованих навігаційних даних.

Слід чітко розмежовувати математичну архітектуру стеганосистеми та критерій живучості MASS як кіберфізичного об'єкта. Безперечно, з погляду фундаментальної теоретичної бази стеганографії, принципу Кірхгоффа, критерій стійкості до відновлення ключа K_4 є детермінуючим: його компрометація веде до повної компрометації стеганосистеми. Проте в задачах забезпечення безпеки автономного судноплавства саме K_3 виступає найважливішим критерієм. Це обумовлено тим, що компрометація K_4 є інформаційною подією (отримання ключа порушником), яка сама по собі ще не порушує рух, стан судна. Натомість руйнування імітостійкості (K_3) є безпосереднім деструктивним впливом (нав'язування хибних команд), що миттєво трансформує інформаційну атаку у фізичну аварію.

Для забезпечення об'єктивності вибору найпріоритнішого критерію стійкості стеганосистеми для MASS слід розмежовувати фізичні наслідки, спричинені руйнуванням прихованого повідомлення (K_5), та наслідки від порушення імітостійкості (K_3). Порушення K_5 відповідно до вимог регуляторних стандартів призведе, як правило, до ініціалізації протоколу безпечної аварійної зупинки, що мінімізує ризики активного маневрування. Натомість порушення імітостійкості (K_3) призводить до прихованого нав'язування хибної інформації. Як наслідок, MASS продовжує активний рух за хибною траєкторією, сприймаючи деструктивні команди як легітимні. Відтак, якщо порушення K_5 веде до пасивної відмови системи, то компрометація K_3 є тригером активного перехоплення управління об'єктом.

Таким чином, саме K_3 є найважливішим критерієм стійкості стеганосистеми з погляду наслідків для живучості MASS.

Визначення критерію імітостійкості (K_3) як найпріоритетнішого ніяк не заперечує необхідності комплексного забезпечення всієї множини вимог безпеки $K = \{K_1, K_2, K_3, K_4, K_5\}$ для MASS. Проте в умовах проєктування, синтезу складних кіберфізичних систем, що функціонують у реальному часі під дією жорстких ресурсних обмежень (обчислювальна потужність борту, пропускна здатність супутникового каналу тощо), виникає задача багатокритеріальної оптимізації. Наявність найпріоритетнішого критерію дозволяє визначити K_3 як головне математичне обмеження при синтезі стеганосистем. Це означає, що при розрахунку параметрів стеганосистеми «математичний запас» стійкості імітозахисту закладається як найвищий. Першочергове забезпечення K_3 ще на рівні моделі знижує ймовірність переходу інформаційної атаки у фізичну аварію MASS.

4. Нові специфічні критерії стійкості для MASS та їхній антагонізм

Специфіка автономного судноплавства як середовища застосування стеганосистеми визначається насамперед характером інформаційних потоків, у які вбудовується прихована інформація. На відміну від класичних сценаріїв застосування стеганографії, де вибір контейнера відносно вільний (при цьому контейнер може бути як фіксований, так і потоковий), а часові характеристики передачі стіганоповідомлення не завжди є критично обмеженими, стеганосистема у складі MASS функціонує в умовах жорстко детермінованого часового середовища. Протоколи управління автономним судном передбачають суворо регламентовані інтервали передачі, відхилення яких є штатним ознакою позаштатної ситуації і негайно фіксується системами моніторингу.

Безумовно, у реальних умовах експлуатації MASS (особливо в зовнішніх супутникових каналах зв'язку) присутнє коливання часової затримки пакетів даних під час їх передачі через мережу - природний часовий джиттер (jitter), обумовлений завадами, специфікою середовища поширення хвиль тощо. Проте межі цього джиттера чітко детерміновані. Будь-яке мікро-відхилення від штатного часового профілю (зокрема викликане алгоритмами стеганографічного вбудовування) миттєво фіксується системами моніторингу як аномальна затримка.

Таким чином, для забезпечення невиявлення прихованого (стеганографічного) каналу додаткова затримка, що вноситься процесом стеганоперетворення, повинна бути меншою за природний джиттер мережі. Інакше противник, що здійснює моніторинг каналу зв'язку автономного судна, отримує можливість виявлення каналу стеганографії не через традиційний стеганаліз вмісту контенту (K_1), а через аналіз часових характеристик трафіку - атака, принципово відмінна від передбачених існуючими критеріями стійкості та специфічна саме для середовища реального часу з жорсткою часовою регламентацією.

Таким чином, для стеганосистеми, що використовується в архітектурі MASS, необхідним є ще один специфічний критерій стійкості, що далі

позначається як K_6 , критерій хронометричної стійкості: спроможність стеганосистеми здійснювати стеганоперетворення наявного контейнера в межах визначеного часового регламенту технологічного процесу, за якої додаткові часові витрати не перевищують межі природного коливання часової затримки в мережі й не створюють часових аномалій для систем моніторингу.

Розглянемо зв'язок введеного критерію K_6 з критерієм K_3 , що має найвищий пріоритет для стеганосистеми в умовах MASS, з позицій синтезу стеганосистеми.

Для досягнення високої імітостійкості (K_3) стеганосистеми природнім, як правило, є застосування складних математичних перетворень, додаткових перевірок при організації прихованого каналу, але кожен додатковий крок збільшує час обробки контенту. Критерій K_6 ставить часові межі. «Нарощування стійкості» відповідно з K_3 може відбуватися лише доти, доки відповідний стеганоалгоритм укладається у відведений часовий ліміт, не створюючи затримок у судовому трафіку, тобто K_6 тут виступає «регулятором і цензором» для математичних методів, якими забезпечується K_3 . Тут потрібно відшукати такий математичний баланс, де висока імітостійкість повністю задовольняє вимогу хронометричної стійкості. Таким чином, можна вважати, що критерій K_6 виступає визначальним інженерним «фільтром» при виборі математичного апарату для забезпечення K_3 : в стеганосистемі процес, що забезпечує імітостійкість, повинен мати незначну обчислювальну складність.

Треба зазначити, що забезпечення незначної обчислювальної складності стеганосистеми є загальною проблемою сучасної стеганографії, оскільки сьогодні на практиці все частіше використовуються потокові контейнери в режимі реального часу, але для стеганосистеми, що є складовою СЗІ MASS, ця проблема стає ключовою. Для класичної стеганографії, основною задачею якої є збереження в тайні наявності прихованого повідомлення, часто в якості основного розглядається критерій K_1 - стійкість до стеганоаналізу. Але, якщо розглянути детально умови застосування стеганосистеми в межах СЗІ MASS, то стає очевидним, що жорсткі часові обмеження можуть зробити класичний стеганоаналіз для порушника не просто важко здійсненим, а нездійсненим взагалі: в умовах потокового контейнера в реальному часі порушник з великою ймовірністю не зможе проводити класичні стеганоаналітичні атаки, за виключенням суб'єктивної, стійкість до якої вимагає лише забезпечення надійності сприйняття стеганоповідомлення, яка передбачає, що порожній контейнер і стеганоповідомлення є такими, що візуально не розрізняються. Це говорить про ще одну особливість для стеганосистеми MASS при її практичному застосуванні: **забезпечення стійкості до стеганоаналізу тут губить свою пріоритетність**, як би це не здавалося безглуздом на перший погляд.

Класична стеганографія розподіляє всі контейнери на випадкові, обрані та нав'язувані. Для будь-якої стеганосистеми для найбільш ефективного забезпечення вимог до неї пріоритетним є обраний контейнер. Пріоритетність вибору контейнера обумовлена тим, що його внутрішня структура та статистичні властивості є одним з обмежувальних факторів для всієї стеганосистеми. Для врахування цього в структурну схему стеганосистеми введено блок «Врахування особливості контейнера» (рис.1), хоча цей блок не є обов'язковим, дозволяючи організацію прихованого каналу зв'язку з випадковим чи нав'язаним контейнером. Взагалі природа контейнера (ентропія, ступінь текстурності тощо) визначає граничну пропускну спроможність формованого каналу, вибір математичного апарату для вбудовування даних, стійкість до стеганоаналізу відповідного стеганоповідомлення. Взагалі існує можливість так обрати контейнер, що у відповідності з використанням секретним ключем він вже буде містити в своєму складі додаткову інформацію (ДІ) – цифрову послідовність, що є результатом роботи прекодера стеганосистеми (рис.1) без будь-яких збурень контейнера під час стеганоперетворення. Це дає можливість отримання стеганоповідомлення, абсолютно стійкого до стеганоаналізу. Такий висновок є сьогодні таким, що заставляє фахівців все більше приділяти уваги методам адаптивної та селективної стеганографії в процесі отримання стеганоповідомлення. Але як вибір контейнера, так і селекція його підобластей, найбільш придатних для вбудови ДІ, вимагає витрати додаткового часу, що, як вже обговорювалося вище, для стеганосистеми в складі СЗІ MASS є небажаним/неможливим. Тому тут найбільш придатним для вибору контейнера може стати його генерація за допомогою ШІ з врахуванням використовуваного стеганоключа і тої керуючої інформації, що передається приховано, наприклад, для визначеності, деяких зображень, що є специфічними саме для MASS.

На сьогоднішній день тут є значна ймовірність отримання як зображення ідеального варіанту з точки зору вбудованої ДІ, але далеко не ідеального з точки зору семантичного контексту, наприклад, наявність тіні від предмета, спрямованої в неприродний бік. Аналіз такого зображення практично миттєво видасть процес його штучної генерації, навіть без використання будь-яких спеціальних програмних засобів (на що, очевидно в умовах експлуатації MASS немає часу), що може розглядатися як опосередкований показник на стеганоперетворення. Справді, навіть інакше на автономному судні генерувати якісь неприродні картини.

Ця атака на семантичну правдоподібність контенту не має аналогів у класичній стеганографії, це абсолютно новий вид атаки, зобов'язаний своєю появою наступним факторам:

- поява та швидке розширення ШІ-генерації інформаційних контентів, які широко використовуються як стеганографічні контейнери останнім часом – цифрові зображення, відео;
- розробка класичних принципів стеганоаналізу відбувалася до епохи впровадження ШІ у всі сфери людської діяльності.

Таким чином, необхідність семантичного аналізу інформаційного контенту породжена саме розвитком технологій ШІ.

Усе сказане вище призводить до необхідності введення ще одного критерію стійкості стеганосистеми в умовах її застосування як складової частини СЗІ MASS – критерію K_7 , що визначається як стійкість до контекстного аналізу: неможливість виявлення факту існування каналу стеганографії через аналіз відповідності контейнера/стеганоповідомлення штатним процесам функціонування системи, у складі якої використовується стеганосистема, природним фізичним явищам (для ЦЗ, ЦВ).

Введення K_7 не змінює пріоритетів існуючих критеріїв стійкості стеганосистеми у складі СЗІ MASS: K_3 зберігає тут найвищий пріоритет, оскільки його порушення безпосередньо тягне за собою виконання судном команд противника з прямими фізичними наслідками, тоді як незабезпечення K_7 є лише передумовою для (можливих) наступних атак.

Остаточна сукупність критеріїв стійкості стеганосистеми у складі СЗІ MASS представлені на рис.4.2.

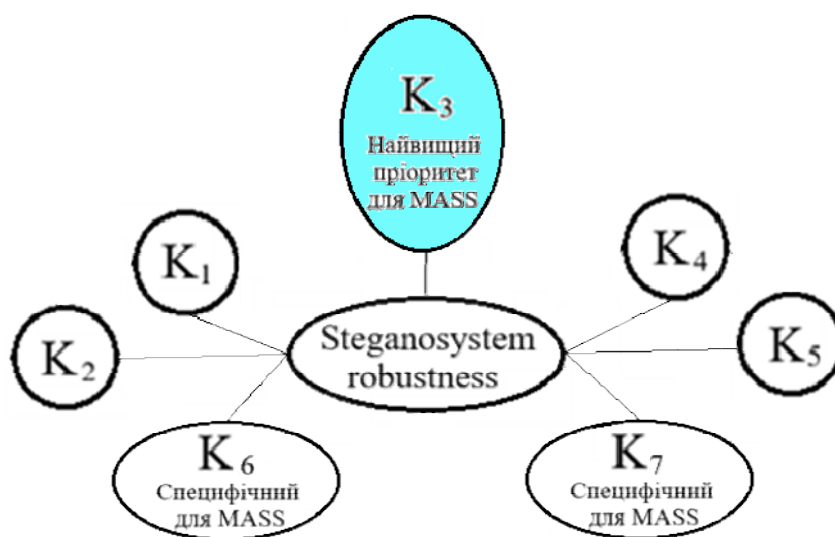


Рис.4.2. Сукупність критеріїв стійкості для стеганосистеми, що є складовою СЗІ MASS

Введення K_6 і K_7 у систему критеріїв стійкості стеганосистеми має важливе практичне значення для проектування стеганосистеми у складі системи захисту інформації MASS. Існуючі до цього моменту критерії $K_1, \dots, K_5$ не містять явних вимог ні до часових характеристик реалізації стеганографічного перетворення, ні до контекстної правдоподібності контейнера/стеганоповідомлення в умовах конкретної галузі застосування. Це означає, що стеганосистема, що формально задовольняє всім існуючим раніше критеріям стійкості, може виявитися вразливою по відношенню до атак, специфічних саме для MASS - часового аналізу трафіку та контекстного аналізу. Тут слід зазначити, що для сучасної стеганосистеми

критерій K_7 є актуальним у будь-якій сфері її застосування, з урахуванням широкого використання сьогодні інформаційних контентів, згенерованих ШІ, однак для MASS цей критерій має виключно важливе значення через виключення можливості застосування адаптивної, селективної стеганографії, про що йшлося вище, що в цьому розумінні робить його специфічним для MASS.

Новий критерій K_6 зобов'язує проєктувальника явно враховувати часові характеристики стеганографічного перетворення та забезпечувати їх відповідність штатним характеристикам протоколів управління судном - вимога, яка без явної наявності K_6 могла бути втраченою при проєктуванні стеганосистеми. Критерій K_7 зобов'язує проєктувальника стеганосистеми забезпечити явно можливість оцінки контекстної правдоподібності обраного контейнера (отриманого стеганоповідомлення) стосовно умов функціонування MASS, виключаючи/обмежуючи, зокрема, використання ШІ-генерованих зображень, семантичні артефакти яких роблять факт стеганографічного каналу таким, що виводиться з контексту без будь-якого стеганоаналізу вмісту.

Таким чином, K_6 і K_7 трансформують неявні вразливості стеганосистеми в MASS у явні вимоги при її розробці, що є необхідною умовою створення дійсно стійкої системи захисту каналу керування автономним судном.

Між введеними критеріями K_6 і K_7 існує певний «антагонізм»: забезпечення контекстної правдоподібності контейнера (K_7) при використанні ШІ-генерації передбачає перевірку семантичної коректності згенерованого зображення/відео перед його передачею - операцію, яка потребує додаткових обчислювальних витрат. Однак будь-яка додаткова затримка вступає у суперечність із вимогою K_6 . Таким чином, чим ретельніше перевірка контекстної правдоподібності, тим вища ймовірність детектування каналу стеганографії через часовий аналіз, і навпаки. Цей антагонізм є специфічним для MASS і не має аналога у класичній теорії стеганографії. Виявлена суперечність між K_6 і K_7 вимагає окремого врахування під час проєктування стеганосистеми для MASS.

Таким чином, продемонстрований якісний перехід від загальних принципів захисту кіберфізичних систем до прикладної реальності на прикладі морських автономних надводних суден.

Очевидно, що у кіберфізичних контурах управління MASS цифрова стійкість не є «офісною надбудовою», вона є первинною умовою існування об'єкта. Будь-який кіберзбій у цій системі неминуче трансформується у фізичну аварію. Через відсутність екіпажу на борту та критичну залежність від берегового центру керування, захист каналу зв'язку вимагає не просто шифрування, а повного приховування самого факту комунікації за допомогою стеганографічних систем. Традиційні критерії стійкості стеганосистеми тут виявилися недостатніми. У реаліях MASS абсолютним

пріоритетом стає імітостійкість (K_3), оскільки саме її порушення призводить до прихованого перехоплення управління судном зловмисником.

Жорсткі часові рамки та використання ШІ-контенту сформували нові вектори атак, що зумовило необхідність введення критерію хронометричної стійкості (K_6) та стійкості до контекстного аналізу (K_7). Виявлений конфлікт між часовою невиразністю трафіку (K_6) та семантичною правдоподібністю контейнера (K_7) доводить, що проєктування захисту для сучасних КФС вимагає пошуку тонких компромісів, а не механічного копіювання рішень.

Ефективний захист кіберфізичних систем нового покоління вимагає створення інтегрованих систем безпеки, які враховують обмеження ресурсів, детермінізм реального часу та специфічні фізичні наслідки інформаційних атак.

Питання

1. До яких наслідків для флоту призвів консерватизм морської галузі та сприйняття кібербезпеки як «офісного» ІТ-завдання? Пояснити.
2. Чому стандартні шаблони корпоративного ІТ-захисту виявляються недієздатними в умовах автономного судноплавства?
3. Як перехід до вищих рівнів автономності трансформує звичайне судно у кіберфізичну систему і чому фізична безпека тут стає похідною від інформаційної?
4. Яку роль у забезпеченні живучості судна відіграє Міжнародний кодекс безпеки для MASS (MASS Code від ІМО), що набув чинності з липня 2026 року?
5. Чому існуючі базові резолюції та циркуляри ІМО (такі як MSC.428(98) чи MSC-FAL.1/Circ.3/Rev.2) не вирішують проблему кібербезпеки у реальному часі?
6. У чому полягає сутність «методологічного вакууму», який виник у морській галузі?
7. Чому стеганографія займає особливе місце серед усіх напрямів кібербезпеки MASS?
8. Яке критичне значення має береговий центр дистанційного керування (ROC) у ситуації повної відсутності екіпажу на борту під час нештатних ситуацій?
9. Чому для захисту каналу управління в MASS недостатньо лише звичайного шифрування, а вимагається приховування самого факту комунікації?
10. Які п'ять базових критеріїв стійкості входять до класичної множини критеріїв стеганосистеми і в чому їх загальна суть?
11. Чому в контурі управління MASS серед усіх класичних критеріїв найвищий пріоритет має саме імітостійкість?
12. Чим відрізняються наслідки порушення імітостійкості від порушення стійкості до відновлення ключа або атак проти вбудованого повідомлення?

13. Що таке хронометрична стійкість і чому перевищення допустимого часового регламенту (наприклад, понад 50 мс) призводить до переведення судна в аварійний режим?
14. Чому в умовах використання штучного інтелекту для генерації контенту виникає необхідність введення критерію стійкості до контекстного аналізу?
15. У чому полягає специфічний для MASS антагонізм між критеріями K_6 та K_7 , і чому він вимагає від проєктувальника свідомого компромісу?

Література

Базова

1. Інтегрована методологія виявлення, аналізу та нейтралізації загроз у сфері національної безпеки і оборони: монографія / С.В.Мілевський, С.С.Погасій, О.А.Лаптев, В.А.Савченко, А.Г.Салій, А.А.Кобозєва, П.В.Жук, Є.О.Меленті. – Харків: НТУ «ХПІ»; Львів: «Новий Світ-2000», 2026. – 424 с.
2. Кіберфізичні системи: технології збору даних [Текст]: монографія - О.Ю. Бочкарьов, В.А. Голембо, Я.С. Парамуд, В.О. Яцук. За редакцією професора А. О. Мельника. Львів: “Магнолія 2006”, 2024. – 176 с.
https://magnolia.lviv.ua/wp-content/uploads/2024/04/Kiiberfizyka-knyha2_zmist.pdf
3. Технології сучасних кібер-фізичних систем: Навчальний посібник [Електронний ресурс]: навч. посіб. для студ. спеціальності 151 «Автоматизація та комп’ютерно-інтегровані технології», освітньо-професійна програма «Автоматизація та комп’ютерно-інтегровані технології кібер-енергетичних систем»; укладач: Ю.Є. Грудзинський. – Електронні текстові дані (1 файл: 14,8 МБ). – Київ : КПІ ім. Ігоря Сікорського, 2020. – 327 с.

Допоміжна

1. Lee E. Cyber-Physical Systems – Are Computing Foundations Adequate? NSF Workshop on Cyber-Physical Systems: Research Motivation, Techniques and Roadmap, Austin, TX (2006).
2. [NIST SP 1500-201. Framework for Cyber-Physical Systems: Volume 1, Overview. Version 1.0. National Institute of Standards and Technology, 2017. 79 p.](#)
3. [Cyber-Physical European Roadmap & Strategy. URL: https://digital-strategy.ec.europa.eu/en/news/cyber-physical-european-roadmap-strategy](#) (дата звернення: 20.07.2025).
4. [Puliafito A. et al. Smart Cities of the Future as Cyber Physical Systems: Challenges and Enabling Technologies. Sensors. 2021. Vol. 21, No. 10. Article No. 3349.](#)
5. [Ryalat M., ElMoaqet H., AlFaouri M. Design of a Smart Factory Based on Cyber-Physical Systems and Internet of Things towards Industry 4.0. Applied Sciences. 2023. Vol. 13, No. 4. Article No. 2156.](#)
6. [Nair M.M., Tyagi A.K., Goyal R. Medical Cyber Physical Systems and Its Issues. Procedia Computer Science. 2019. Vol. 165. P. 647-655.](#)
7. Yu X., Xue Y. Smart Grids: A Cyber-Physical Systems Perspective. *Proceedings of the IEEE*. 2016. Vol. 104, No. 5. P. 1058-1070.
8. Khan R. et al. Threat Analysis of BlackEnergy Malware for Synchrophasor based Real-time Control and Monitoring in Smart Grid. Proceedings of the 4th International Symposium for ICS & SCADA Cyber Security Research 2016 (ICS-CSR), pp. 53-63.

9. [Ahmadi A., Cherifi C., Cheutet V., Ouzrout Y. A Review of CPS 5 Components Architecture for Manufacturing Based on Standards. Proceedings of the 11th IEEE International Conference on Software, Knowledge, Information Management and Applications \(SKIMA 2017\), Dec 2017, Colombo, Sri Lanka. 6 p.](#)
10. Casola V., De Benedictis A., Mazzocca C., Montanari R. Designing Secure and Resilient Cyber-Physical Systems: A Model-Based Moving Target Defense Approach. *IEEE Transactions on Emerging Topics in Computing*. 2024. Vol. 12, No. 2, P. 631-642.
11. [Wiesner S. et al. Requirements Engineering for Cyber-Physical Systems. URL:
https://doi.org/10.1007/978-3-662-44739-0_35 \(дата звернення: 20.07.2025\).](#)
12. [Mandrioli C., Carlsson M., Maggio M. Testing Abstractions for Cyber-Physical Control Systems. ACM Transactions on Software Engineering and Methodology. 2024. Vol. 33, No. 1. Article 18.](#)
13. [Rolofs G., Wilking F., Goetz S., Wartzack S. Integrating Digital Twins and Cyber-Physical Systems for Flexible Energy Management in Manufacturing Facilities: A Conceptual Framework. Electronics. 2024. Vol. 13, No. 24. Article No. 4964.](#)
14. Wang C. Zigbee Network Protocols and Applications. Auerbach Pub, 2014. 353 p.
15. Townsend K. Getting Started with Bluetooth Low Energy: Tools and Techniques for Low-Power Networking. O'Reilly Media, 2014. 178 p.
16. Unwala I., Taqvi Z., Lu J. Thread: An IoT Protocol. Proceedings of the 2018 IEEE Green Technologies Conference (GreenTech), Austin, TX, USA, 2018, pp. 161-167.
17. Paetz C. Z-Wave Essentials. CreateSpace Independent Publishing Platform, 2018. 310 p.
18. Biswajit Paul. An Overview of LoRaWAN. *WSEAS Transactions on Communications*. 2020. Vol. 19. P. 231-239.
19. Naeem M. et al. Modelling and Analysis of a Sigfox-Based IoT Network Using uppaalsmc. *IEEE Sensors Journal*. 2023. Vol. 23, No. 10. P. 10577-10587.
20. [Modicon Modbus Protocol Reference Guide \(PI-MBUS-300 Rev. J\). URL:
https://www.modbus.org/docs/PI_MBUS_300.pdf \(дата звернення: 20.07.2025\).](#)
21. Mahnke W., Leitner S.-H., Damm M. OPC Unified Architecture. Springer, 2009. 362 p.
22. Heath S. Embedded Systems Design. Newnes, 2002. 430 p.
23. Kumar N.S., Saravanan M., Jeevananthan S. Microprocessors and Microcontrollers. Oxford University Press, 2011. 765 p.
24. Karimipour H. Security of Cyber-Physical Systems: Vulnerability and Impact. Springer, 2020. 333 p.
25. Xing W., Shen J. Security Control of Cyber-Physical Systems under Cyber Attacks: A Survey. *Sensors*. 2024. Vol. 24, No. 12. Article No. 3815.
26. Shostack A. Threat Modeling: Designing for Security. Wiley, 2014. 624 p.

27. Ji X., Yu H., Fan G., Fu W. Attack-defense trees based cyber security analysis for CPSs. Proceedings of the 2016 17th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), Shanghai, China, 2016, pp. 693-698.
28. Celikel Cankaya E. Security and Privacy in Three States of Information. URL: <http://dx.doi.org/10.5772/intechopen.91610> (дата звернення: 20.07.2025).
29. Shepherd C., Markantonakis K. Trusted Execution Environments. Springer, 2024. 224 p.
30. Munjal K., Bhatia R. A systematic review of homomorphic encryption and its contributions in healthcare industry. *Complex & Intelligent Systems*. 2023. Vol. 9. P. 3759–3786.
31. Mukherjee A. The Complete Guide to Defense in Depth. Packt Publishing, 2024. 298 p.
32. Mitchell R., Chen I.-R. A survey of intrusion detection techniques for cyber-physical systems. *ACM Computing Surveys*. 2014. Vol. 46, No. 4. Article 55.
33. Siddique M.S. et al. An intelligent intrusion detection system for cyber-physical systems using GAN-LSTM networks. *Franklin Open*. 2025. Vol. 11. 100281.
34. Foreman J. et al. Detection of Hacker Intention Using Deep Packet Inspection. *Journal of Cybersecurity and Privacy*. 2024. Vol. 4, No. 4. P. 794-804.
35. Fausto A. et al. Toward the Integration of Cyber and Physical Security Monitoring Systems for Critical Infrastructures. *Sensors*. 2021. Vol. 21, No. 21. Article No. 6970.
36. Gonzalez-Granadillo G., Diaz R., Caubet J., Garcia-Milà I. CLAP: A Cross-Layer Analytic Platform for the Correlation of Cyber and Physical Security Events Affecting Water Critical Infrastructures. *Journal of Cybersecurity and Privacy*. 2021. Vol. 1, No. 2. P. 365-386.
37. Papp D., Ma Z., Buttyan L. Embedded systems security: Threats, vulnerabilities, and attack taxonomy. Proceedings of the 2015 13th Annual Conference on Privacy, Security and Trust (PST), Izmir, Turkey, 2015, pp. 145-152.
38. Vålja M., Korman M., Lagerström R.. 2017. A Study on Software Vulnerabilities and Weaknesses of Embedded Systems in Power Networks. Proceedings of the 2nd Workshop on Cyber-Physical Security and Resilience in Smart Grids (CPSR-SG'17). New York, NY, USA, 2017, pp. 47–52.
39. ISA/IEC 62443 Series of Standards. URL: <https://www.isa.org/standards-and-publications/isa-standards/isa-iec-62443-series-of-standards> (дата звернення: 20.07.2025).

Інтернет ресурси

1. Директива Європейського Парламенту і Ради (ЄС) 2022/2555 від 14 грудня 2022 року про заходи для високого спільного рівня кібербезпеки на всій території Союзу, внесення змін до Регламенту (ЄС) № 910/2014 та Директиви (ЄС) 2018/1972 та скасування Директиви (ЄС) 2016/1148 (Директива NIS 2). URL: https://zakon.rada.gov.ua/laws/show/9a3_001-22

(дата звернення: 20.06.2025)

2. Міжнародний кодекс безпеки морських автономних надводних суден (MASS Code) <https://www.imo.org/ru/mediacentre/pressbriefings/pages/imo-adopts-mass-code.aspx>