DOI:https://doi.org/10.26565/2304-6201-2024-63-04

УДК (UDC) 004

Кіреєнко Олександр Володимирович *аспірант,асистент кафедри інформаційної безпеки Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського", Берестейський проспект, 37, Київ, 03056*
e-mail: kirealex12@gmail.com
*OrcidID:*0000-0001-9184-6738

Захист інформації в умовах обмежених енергоресурсів

Актуальність: Захист інформації та інформаційної системи, в якій дана інформація обробляється та зберігається, є складною задачею навіть за оптимальних умов. Накладання будь-яких обмежень призводить до появи нових проблем. Обмеження енергоресурсів змушує нас змінити підхід до захисту інформації, відмовитися від деяких рішень, а також змінити спосіб використання інформаційної системи. в роботі основна увага приділяється системам, що використовують мережеве з'єднання. Мережеве обладнання розглядається як частина системи і потребує живлення.Проблема обмежених енергоресурсів розглядається на двох рівнях – в межах однієї системи та глобально, коли інформаційні системи конкурують за обмежені ресурси. Порушнику відомо про функціонування системи в умовах обмежених енергоресурсів, але в загальному випадку не відомо наскільки критичною є їх нестача.
Мета: побудова моделі захисту інформації із заданими обмеженнями, вибірка задач із високим співвідношенням винагороди за виконаня до витрат енерго ресурсів, створення резерву енергії для його подальшого використання в наступний період функціонування системи, оцінка ризику перевантаження енергомережі.
Методи дослідження: теорія графів, теорія ігор, методи Фішберна, Уейна для вибору кращого проекту
Результати: побудовано модель і отримано 2 стратегії розподілу енергоресурсів
Висновки: проблема нестачі енергоресурсів в загальному випадку вирішується за рахунок надлишковості обчислювальних ресурсів та пам'яті. Проведення атаки на інформаційну систему як і її припинення впливають на розподіл задач, що надходять до системи від законних користувачів. Вплив на вартість енергоресурсів можливий, але виходить за межі моделі.

Ключові слова:енергоресурси, автономне джерело живлення, мережеве обладнання, мобільні пристрої, планування задач, резервне копіювання

Як цитувати: Кіреєнко О. В. Захист інформації в умовах обмежених енергоресурсів. *Вісник Харківського національного університету імені В. Н. Каразіна, серія Математичне моделювання. Інформаційні технології. Автоматизовані системи управління*. 2024. вип. 63. С.38-50. https://doi.org/10.26565/2304-6201-2024-63-04

How to quote: O. Kireienko,"Information protection in the conditions of limited energy resources", *Bulletin of V. N. Karazin Kharkiv National University, series Mathematical modelling. Information technology. Automated control systems*, vol. 63, pp. 38-50, 2024.https://doi.org/10.26565/2304-6201-2024-63-04

Вступ

Задачу захисту інформації в умовах обмежених енергоресурсів можна розбити на 3 етапи або підзадачі. За основу беруть проект системи, що відповідає тій самій предметній області, обслуговує подібні бізнес процеси, але без обмежень на енергоресурси. Більшість атак, що є стандартними для такої системи, можуть бути спрямовані і на систему з обмеженими енергоресурсами. На 2 етапі ми можемо відсіяти атаки, які стають неможливими через обмеження енергоресурсів. найчастіше такі атаки проводяться внутрішніми порушниками-інсайдерами, які повинні взаємодіяти із системою, враховуючи нові обмеження. Останній крок – розглянути специфічні атаки, коли порушник цілеспрямовано використовує нестачу енергоресурсів інформаційної системи для проведення атаки.

На даний момент основним підходом до вирішення проблеми захисту інформації в умовах обмежених енергоресурсів є вибір проектного рішення системи, при якому обмеження знімаються, тобто придбання в достатній кількості автономних джерел живлення. Основним

недоліком даного підходу є припущення про відсутність дефіциту автономних джерел живлення на ринку. Стихійні лиха та війни призводять до швидкого зростання кількості систем, яким знадобиться працювати в заданих обмеженнях і зростанню попиту на автономні джерела живлення. Саме тому необхідно розглянути альтернативні проекти, при яких система зможе виконувати поставлені задачі. Опосередковано проблема обмеження енергоресурсів пов'язана із питаннями використання альтернативної операційної системи (dualboot) [1], прискорення завантаження операційної системи [2], розрахунку споживання електроенергії центральним процесором (CPU) [3], та забезпечення вимог інформаційної безпеки в нестабільних мережах [6]. Теоретичною базою для розподілу енергоресурсів є "гра із забрудненням" із теорії ігор [4]. До технічних аспектів відноситься використання мобільних пристроїв як альтернативи персональним компютерам [5], прогнозування розгалужень при виконанні коду [7], використання транзакцій і збереження їх результатів в пам'яті [8] та ін.

Метою статті є розробка моделі захисту інформації в умовах обмежених енергоресурсів, та вибір сценаріїв захисту системи відповідно до такої моделі.

1. Види обмежень

Будь-яка інформаційна система використовує джерело живлення. Система може отримувати живлення від мережі або від акумулятору, який в свою чергу заряджається від мережі. Існує декілька обмежень пов'язаних із функціонуванням системи в умовах обмежений енергоресурсів.

Першим обмеженням є організаційне. Це самообмеження, що накладається користувачами на систему, або самою системою на себе, і пов'язане воно із типом задач, які взагалі будуть виконуватися системою. Користувачі, оцінивши ризик відключення живлення в довільний момент часу можуть відмовитися від виконання задач, які потребують наявності живлення протягом всієї задачі. Також користувачі можуть оцінити час виконання задачі і порівняти його із часом автономного живлення системи від акумулятора або генератора, або від мережі, за умови, що у користувачів є надійний графік відключень електроенергії. В таких випадках користувачі самі вирішують, які задачі будуть виконуватися в системі, та порядок дій, якщо живлення все ж зникне. Дані задачі можуть бути як прикладними, коли система використовується за призначенням, так і службовими, що виконуються для підтримки системи в стабільному та захищеному стані.

Другим обмеженням є обмеження обчислювальних ресурсів системи. Задачі, що потребують більших обчислювальних потужностей, витрачатимуть більше енергії, так як цю енергію споживає центральний процесор компютера, і також система охолодження. Для систем, що використовують автономне джерело живлення, виконання важких обчислювальних задач призведе до швидшого витрачання наявної електроенергії. В той же час, при наявному графіку відключень матиме сенс виконання таких задач в певні інтервали часу, коли електроенергія точно є. Це в свою чергу створює вразливість в системі, так як порушнику буде відомо, коли саме виконуватимуться відповідні задачі, а значить він зможе вибрати оптимальний для проведення атаки час.

Третім обмеженням є обмеження, що пов'язані із типом пристрою, який використовується для виконання всіх цих задач. В умовах обмежених енергоресурсів користувачі будуть полягатися на мобільні пристрої. Захист мобільних пристроїв є окремим напрямком інформаційної безпеки, але в даному випадку ми не можемо використовувати в повному обсязі рекомендації щодо захисту мобільних пристроїв тому що ми розглядаємо ситуацію, коли мобільні пристрої використовуються не за своїм основним призначенням, а саме як тимчасова заміна персональним комп'ютерам. Це означає, що додатки мобільних пристроїв можуть почати конфліктувати із додатками, які повинні взяти на себе виконання задач персонального комп'ютера (при цьому конфліктують вони і за ресурси і за час). Також потрібно враховувати, що операційні системи мобільних пристроїв відрізняються від операційних систем персональних комп'ютерів, і що ми не можемо в загальному випадку запустити додаток, що працював на персональному комп'ютері. Нам знадобляться програми-аналоги, емулятори, клієнт-серверна архітектура з доступом до хмарних сервісів.

Четвертим обмеженням є обмеження пов'язане із інтернет-з'єднанням. Відсутність живлення означає, що наша система може вимушено переходити в автономний режим роботи. Навіть якщо сам пристрій може працювати, використовуючи акумуляторну батарею, мережеве обладнання

може в цей час бути недоступним, а тому виконання задач потрібно планувати і з розрахунком на нестабільне інтернет з'єднання.

Наслідком четвертого обмеження є проблема отримання інформації із всесвітньої мережі. За відсутності живлення ми не зможемо використовувати ресурси, що доступні лише віддалено. Як і у випадку із плануванням задач у відповідності із графіком відключень електроенергії, порушник може використати цю інформацію для проведення атак, що спрямовані на перехоплення трафіку, так як порушнику відомо, коли саме система матиме зв'язок з мережею, а значить саме в цей час відбуватиметься передача інформації (наприклад результати виконання прикладних задач). Сам факт обмеження часу для обмінуінформацією робить будь-яку систему більш вразливою до атаки на відмову в обслуговуванні, так як тепер порушнику достатньо підтримувати подібну атаку лише в проміжки часу, коли в системи є живлення.

В наступних розділах дані проблеми будуть розглянуті більш детально.

2.Організаційні обмеження

В умовах обмежених енергоресурсів користувачі системи повинні вирішити, які задачі можна виконувати, а які треба відкласти на потім. Дане рішення залежить від того, в яких саме умовах знаходиться система. Наявність автономного джерела живлення може вказувати на задачі, які можна вирішити протягом часу автономної роботи. Якщо автономного джерела живлення немає, а замість цього відбувається живлення від мережі і є **ризик** відключення електроенергії в деякий момент часу, користувачі повинні будуть планувати виконання задач із урахуванням цього ризику. Ми можемо використовувати моделі із теорії масового обслуговування для прогнозування часу безперебійної роботисистеми. Прикладні та службові задачі необхідно розміщувати між моментами відключення та відновлення живлення. Ввімкнення комп'юера, завантаження його операційної системи займає певний час, який треба віднімати від загального часу, що доступний для виконання будь-яких задач. Так як ми розглядаємо саме випадок із відключенням живлення, ми не можемо використовувати режим гібернації, що дозволив би швидко ввімкнути комп'ютер. Unix подібні операційні системи потребують менше ресурсів і можуть завантажуватися швидше. Час завантаження можна скоротити ще більше, якщо відмовитися від графічного інтерфейсу. Але подібний підхід не є універсальним. Для вирішення деяких прикладних задач нам можуть знадобитися додатки, що доступні лише для ОС windows. Оптимальним варіантом буде використання комп'ютерів, на яких Unix-подібна ОС встановлена як запасна. В цьому випадку для вирішення окремих задач ми можемо під час включення комп'ютера обрати саме Unix подібну ОС, в той час як для інших використовуватимуть ОС Windows або інші подібні ОС.[1]

Так само потрібно пам'ятати і про час для відключення комп'ютера. Для захисту жорсткого диску нам потрібно відключати комп'ютер програмними засобами, тому при наявному графіку відключень електроенергії нам потрібно виділити в кінці періоду функціонування декілька хвилин для збереження результатів роботи та відключення комп'ютера, перш ніж його роботу буде перервано апаратно. Швидкість запуску ОС після завершення роботи можна підвищити, якщо зберегти стан системи в пам'яті, але подібна процедура сама по собі є витратою часу. [2] Таким чином ми можемо скоротити час ввімкнення системи за рахунок збільшення часу для її відключення.

Необхідність виконання задач в задані інтервали часу накладається на інші логічні обмеження щодо часу виконання задач. Якщо нам відомо, що живлення буде з 7 до 10 години ранку, а робочий день починається о 8:30, то в нас буде лише півтори години (без урахування декількох хвилин на ввімкнення та вимкнення системи) для вирішення деяких задач. В найкращому випадку користувачі адаптуються до нового розкладу і будуть працювати із системою саме тоді, коли є живлення. Це можна спостерігати у випадку роботи із персональними комп'ютерами вдома. Змінювати графік роботи на підприємствах значно складніше, тому що в доступний для роботи час, користувачів може просто не бути на робочому місці.

Ще одним організаційним обмеженням буде сумісність деяких задач. Розглянемо наступний приклад

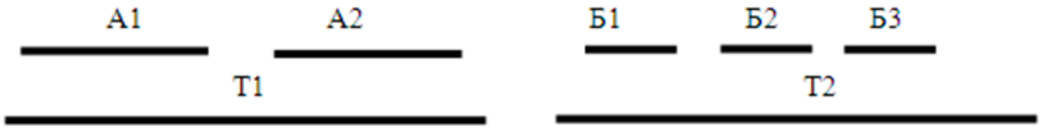

Рисунок 1.А Планування незалежних задач

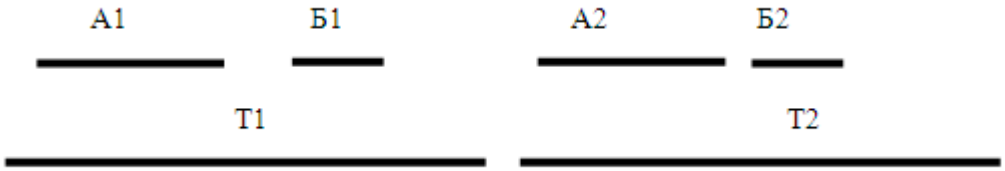

Рисунок 1.Б Планування послыдовних задач

Нехай Т1 та Т2 - періоди наявного живлення, а А1, А2, Б1, Б2, Б3 - періоди виконання деяких задач. На рисунку 1.А задачі згруповані досить оптимально, спочатку дві задачі, які займають приблизно половину часу від доступного нам періода наявного живлення, потім три короткі задачі, які вкладаються в період Т2. Але такий "оптимлальний" варіант планування задач може бути неможливим, якщо задача Б1 повинна використовувати результат виконання задачі А1 (аналогічно для Б2 та А2), який при цьому швидко застаріває. Якщо результат виконання А1 (і А2) втрачає актуальність в проміжок часу між завершенням Т1 та початком Т2, то планування задач необхідно виконувати відповідно до рисунку 1.Б, навіть якщо при цьому в нас з'являється час простою системи при наявному живленні. Так як в деяких випадках ми просто не в змозі змінити логіку бізне спроцесу, який забезпечується нашою інформаційною системою, найкращим рішенням буде виконання деяких додаткових прикладних або службових задач для уникнення часу простою системи.

Із рисунків 1.А та 1.Б також слідує, що сам факт переривання живлення навіть на декілька хвилин може призвести до непропорційно більшого зниження продуктивності інформаційної системи. Тобто ми не можемо оцінювати ефективність інформаційної системи як просте відношення часу наявного живлення (Т1+Т2) до загального часу.

На організаційному рівні всі задачі поділяються на невідкладні (urgenttasks), поточні (currenttasks) та профілактичні (maintenancetasks). $T = \{U\_t, C\_t, M\_t\}$

Невідкладні задачі потрібно виконувати обов'язково в момент їх надходження до системи. Якщо в даний момент система зайнята іншою задачею, що не відноситься до підмножини невідкладних, її потрібно призупинити або відмінити, і почати виконувати невідкладну. Задачу потрібно відмінити навіть якщо вона виконана на 90+% і ми не можемо зберегти результат виконання. Якщо система зайнята іншою невідкладною задачею, і при цьому поступає нова невідкладна задача, для якої не вистачає ресурсів – система вважається скомпрометованою, а відповідний бізнес процес повністю зірваним.

Поточні задачі – це задачі, які повинні бути виконані в день їх надходження (замість однієї 24-годинної доби може бути обраний інший інтервал часу, це не впливає на класифікацію). Поточні задачі повинні бути виконані до деякого дедлайну.

профілактичні задачі – не є обов'язковими для виконання, але їх виконання підвищує стабільність роботи системи. Тривале ігнорування цих задач може призвести до їх переходу до підмножини невідкладних або поточних.

До якої б із підмножин не відносилася задача, їй у відповідність можна поставити у відповідність двовимірний масив (матрицю) розмірності $n \times 3$

$$t_i \Rightarrow \begin{pmatrix} 1,0,0\ldots,1,0 \\ d1,d2,d3,\ldots,d(n-1),dn \\ c1,c2,c3,\ldots,c(n-1),cn \end{pmatrix} \tag{2.1}$$

перший рядок матриці заповнюють одиницями та нулями. Він відповідає за можливість виконання і-тої задачі на кожному із доступних в системі вузлів.

другий рядок відповідає за тривалість (duration) виконання задачі на заданому вузлі. Очевидно, що якщо в першому рядку на відповідній позиції стоїть 0, то час виконання буде невизначеним. Для простоти обчислень можна просто використовувати від'ємне число.
третій рядок відповідає за вартість виконання задачі на цьому вузлі (cost) в розряхунку на одиницю часу. таким чином вартість задачі є добутком елементів в стовпці матриці. нульові значення просто ігноруємо, тому що миттєво виконати задачу не можливо. Значення нульове тому, що в перщому рядку в нас був 0.
в третьому рядку вартість можна вказувати як в грошових одиницях так і в енергії, але треба, щоб в усіх матрицях застосовувалися однакові одиниці вимірювання.

Дані матриці не є константами. Назначення задачі якомусь вузлу означає, що в нього буде зайнята вже деяка частка обчислювальних ресурсів, тому після отримання декількох задач, в першому рядку 1 буде замінено 0 а при завершенні задачі відбувається протилежна операція.

Також не будуть константами і значення в 3 рядку, так як при збільшенні кількості задач на вузлі зростатиме температура ЦП, що може потребувати додаткових витрат на його охолодження.
Недоступним для призначення нових задач вузол також стає тоді, коли в нього закінчуються електроенергія.

Є два підходи до розподілу задач. Перший підхід – це призначення задач на вузол до вичерпання всіх його ресурсів, і лише після цього переходимо до наступного вузла. При такому підході ми мінімізуємо кількість задіяних вузлів. Задіяні вузли можуть витрачати трохи більше енергії через високу завантаженість, але всі інші вузли, на яких відсутні задачі, не витрачатимуть енергію взагалі. Другий підхід – не рівномірний розподіл задач, коли навантаження балансують між усіма вузлами. При першому підході більша імовірність того, що в нас буде вільний від задач вузол, який зможе почати виконувати невідкладну задачу, якщо така надійде. В другому випадку – через активність ВСІХ вузлів системи нам доцільно підтримувати середній рівень завантаженості вузлів за рахунок більшої кількості поточних та профілактичних задач. Другий випадок означає, що імовірність переривання та/або витіснення якоїсь задачі буде близькою до 100%, якщо надійде невідкладна задача, але більша завантаженість усіх вузлів означає, що жоден із них не використовує надмірну кількість енергії на охолодження. також зменшується вікно протягом якого нам може знадобитися перервати чи витіснити задачу для виконання невідкладної задачі. Тобто за перші декілька годин робочого дня ми виконаємо більшість задач, що були заплановані на сьогодні, після чого залишається опрацьовувати лише нові невідкладні задачі. При цьому нам не обов'язково щодня використовувати один і той самий підхід. Ми можемо змінювати підхід в залежності від ситуації.

3.Обмеження обчислювальних ресурсів

Для даного обмеження можливі два сценарії:
- по-перше це може бути обмеження для автономного джерела живлення, коли важкі обчислення призводять до швидшого витрачання енергії[3]
- по-друге це може бути обмеження на всю енергомережу, коли виконання важких обчислень кожним окремим вузлом не є проблемою, але виконання тих же задач деякою критичною кількістю вузлів призведе до відключення живлення всієї системи. Це обмеження є класичним випадком "гри із забруденнням" із теорії ігор. [4] Гра із забрудненням є гіпотетичним сценарієм, в якому гравцями є виробництва, кожне із яких шкодить довкіллю. Окреме виробництво не здатне завдати суттєвої шкоди, так як в довкілля є деякий запас міцності, що дозволяє йому самовідновлюватися. Очевидно, що при перевищенні даного запасу гравці починають завдавати незворотної шкоди довкіллю, що закінчується трагічно для всіх. Метою даного прикладу є демонстрація самообмеження, коли гравці відмовляються від шкідливого виробництва, щоб всі гравці не програли одночасно. Проблема в тому, що відмова від виробництва означає, що даний гравець в даному раунді гри не отримує прибутку, в той час як інші гравці, що вирішують ігнорувати проблеми довкілля, такий прибуток отримують. А за правилами гри перемагає той, хто набрав найбільший прибуток. Парадоксально перемогти може лише той, хто забруднює довкілля, але якщо таких гравців набирається більше деякої кількості, то програють всі. Замість забруднення довкілля в нашому випадку споживання електроенергії від мережі. Прибуток є наслідком виконаних задач, гравцями є вузли системи або її користувачі. І ці вузли/користувачі повинні обрати між вирішенням деякої підмножини задач або взагалі нічого не зробити.

Складність залежить від того, чи є між гравцями зв'язок постійно/перед грою чи зв'язок взагалі відсутній. Чи можуть вони узгодити розклад за яким в кожному раунді гри доступ до електроенергії буде отримувати деякий користувач, чи вони діятимуть егоїстично? У даної проблеми є ще один вимір. Крім вибору вузлів, що виконуватимуть задачі, користувачі також можуть обирати самі задачі, які виконуватимуться в системі. Прикладні задачі можна розглядати як дії, що призводять до забруднення, в той час як службові (резервне копіювання, перевірка антивірусом та ін.) можуть знижувати рівень забруднення. Для отримання максимального прибутку користувачам потрібно збільшити кількість прикладних задач, а зробити це можна лише за рахунок задач службових, тобто користувач може прийняти рішення щодо відключення механізмів захисту. Чим більше механізмів захисту буде відключено, тим більше імовірність успішної атаки порушника, що в свою чергу означає, що прибутку від прикладних задач не буде взагалі (критичний рівень забруднення, який не може витримати довкілля).

Гра із забрудненням є чітко формалізованою, зрозумілою проблемою і в мережі можна знайти безліч програмних реалізацій та моделей. Опишемо відповідну модель тут:

- нам потрібна множина із пелементів $A = \{node_1, node_2, \dots, node\_n\}$
- кожен елемент має визначені для нього імовірності p та q, такі що $p + q = 1$
- для і-того вузла імовірність p_i відповідає за "егоїстичну" стратегію, коли вузол приймає рішення забруднювати (в нашому випадку – виконувати складну та енергозатратну задачу), а q_i – альтруістичну стратегію, коли подібні задачі не виконуються.
- генеруємо випадкове число із діапазону [0,1]. Якщо воно потрапляє в діапазон [0,p] функція даного вузла повертає 1, в протилежному випадку повертає 0
- рахуємо суму відповідей від всіх вузлів і порівнюємо її із пороговим значенням. $\sum_{1=1}^{n} b_i$=B, $B < T$– забруднення не відбулося, $B \geq T$ – забруднення відбулося, всі програли, мережа не витримала.
- Виконуємо прогонку даної моделі декілька разів і рахуємо частку випадків, коли забруднення відбулося до загальної кількості. Чим більше ітерацій тим ближче експериментальне значення likelyhood до теоретичної імовірності того, що забрудення відбудеться і нам доведеться використовувати автономні джерела живлення.

Зауваження: програш у грі із забрудненням ще не означає крах всієї системи. Він вказує лише на необхідність застосування автономних джерел живлення, яких може бути достатньо для підтримки системи, за умови що програші відбуваються не щодня і в нас вистачає часу відновити автономні джерела живлення перед наступним програшем (отримати пальне для генераторів, підзарядити акумуляторні батареє від мережі, тощо)

4. Обмеження типу пристрою

Очевидним вирішенням проблеми із живленням інформаційної системи є використання мобільних пристроїв так як вони мають автономне джерело живлення у вигляді акумуляторної батареї.Але не всі задачі можуть бути перенесені на мобільні пристрої. По-перше всі задачі, які потребують більших запасів енергії ніж доступні в автономному джерелі живлення нам не підходять, так як вони не зможуть бути завершені. Мобільні пристрої мають більш жорсткі обмеження на пам'ять, а також потребують додаткового обладнання для взаємодії із периферійними пристроями. У персонального комп'ютера може бути CD/DVD-rom для роботи з оптичними дисками. Смарфтону або планшету для цього знадобиться спеціальний перехідник з USB. При цьому USB роз'єм зайнятий підключенням CD/DVD-rom вже не можна буде використати для підключення інших пристроїв (наприклад принтеру або сканеру). Звичайно, при наявності всього необхідного обладнання користувач може використовувати CD/DVD-rom та принтер по черзі. Це дозволить виконувати деякі задачі повільніше. Наприклад ми можемо зберегти файл із оптичного диску в пам'ять смартфону, а потім роздрукувати даний файл на

принтері. Для даної послідовності дій нам потрібно, щоб на смартфоні було достатньо пам'яті для тимчасового зберігання файлу. Навіть якщо смартфон підтримує бездротове з'єднання (Wi-Fiабо bluetooth), таке ж з'єднання повинен підтримувати і периферійний пристрій, який ми намагаємося підключити. Бездротове підключення є більш поширеним серед сучасного периферійного обладнання із середнього та вищого цінового сегменту. Дешеве або старе периферійне обладнання можна підключати лише за допомогою дротового з'єднання.

Проблеми можуть виникнути і з програмним забезпеченням. ОС Android була розроблена на основі linux, але це не означає,що ми можемо запускати bash-скрипти на Android. Для роботи з терміналом нам потрібно встановити відповідне програмне забезпечення, і робити це треба завчасно, а не після того, як відключення живлення стало проблемою. Для мобільних пристроїв існує також емулятор MS-DOS, що дозволяє запускати старі виконувані файли. Але працювати із звичайними .exe файлами ми не зможемо. Будь-який важливий для роботи додаток повинен бути замінений його мобільною версією, а за відсутності, такий аналог потрібно написати самостійно, що для більшості користувачів не є прийнятним рішенням даної проблеми. Для вирішення прикладних задач нам може бути достатньо мови програмування QBasic, так як написані на цій мові програми можназапускати в емуляторі MS-DOS. Для інших мов програмування нам знадобиться компілятор і середовище розробки, що здатні працювати автономно, без підключення до інтернету. Онлайн компілятори все ще можуть використовуватися в проміжки часу, коли в нас є живлення мережевого обладнання.

Не можна ігнорувати і проблему сумісності мобільних додатків та додатків, які повинні тимчасово виконувати завдання персонального компютера. Користувач повинен розуміти, що при відключенні живлення, коли персональний комп'ютер не працює, його мобільний пристрій виконує задачі персонального комп'ютера як додаткові до тих, що вже виконувалися на мобільному пристрої в його штатному режимі функціонування. Це означає, що використовуючи мобільний пристрій в цихумовах, навантаження на ньому буде значно більше, бо на ньому виконуються власні задачі та задачі перенесені із ПК. Це підвищене навантаження призводить до швидшого витрачання енергії, а значить скорочує період автономної роботи пристрою. Це також означає, що при більшому ніж в штатному режимі роботи навантаженні, службові програми, що необхідні для захисту інформації можуть бути відключені тому, що на їх підтримку просто не вистачатиме ресурсів.

Також потрібно враховувати, що порушник може використати вразливості мобільного пристрою для доступу до інформації, яка зазвичай обробляється на персональному комп'ютері, до якого у порушника доступу не було.[5]Збільшення кількості задач на будь-якому пристрої робить його більш вразливим до атак на відмову в обслуговуванні (через брак ресурсів), а також підвищує інтерес порушника до даного вузла системи.

5.Обмеження пов'язані з інтернет-з'єднанням

Нестабільне інтернет з'єднання означає, що топологія нашої інформаційної системи змінюється протягом виконання поставлених задач [6]

Всі задачі можна розділити на п'ять основних категорій:

1) задачі, що потребують постійного інтернет з'єднання
2) задачі, що потребують інтернет-з'єднання для ініціалізації
3) задачі, щопотребують інтернет-з'єднання для надсилання результатів
4) задачі, щопотребують інтернет-з'єднання для корекції
5) задачі, що не потребують інтернет-з'єднання

Можливими є і комбінації 2+3, 2+4, 3+4, 2+3+4

Для задач, що потребують постійного інтернет-з'єднання ми повинні використовувати ту ж стратегію, що була описана в розділі **організаційні обмеження.** Тобто ми повинні планувати їх виконання на проміжки часу, коли інтернет-з'єднання гарантовано буде. Переривання інтернет-з'єднання може означати, що задачу потрібно почати виконувати повторно, або що її результат виконання не можна буде застосовувати. В останньому випадку нам потрібно виявити факт переривання інтернет-з'єднання, щоб одразу перервати задачу і не завантажувати нашу систему, так як результат виконання цієї задачі все одно буде незадовільним.

Задачі, щопотребують інтернет-з'єднання для ініціалізації – це задачі, які отримує сервер в клієнт-серверній архітектурі. Для зменшення впливу обмеження живлення на виконання таких задач бажано реалізувати чергу і буфер повідомлень, щоб ми могли прийняти одразу деяку

послідовність таких задач, і виконувати їх в автономному режимі, поки мережеве обладнання не працює через відсутність живлення. Проблема використання черги полягає в тому, що в черзі будуть необов'язкові завдання, які туди потрапили тому, що без ініціалізації їх неможливо запустити, та у рівні впевненості меншому ніж 100% в тому, що виконання цих задач нам знадобиться. Тобто через невизначеність із тривалістю відсутності живлення ми відправляємо на сервер додаткові задачі, які сервер повинен почати виконувати в автономному режимі після основних задач із черги, якщо живлення не буде відновлено. Чим довше буде відсутнім живлення мережевого обладнання, тим більше сценаріїв розвитку подій для серверу ми повинні передбачити, так як набір задач ми можемо відправити лише один раз. Навіть при роботі ізоднією задачею, в ній може бути передбачено розгалуження. Можливість передбачити маршрут, який буде обрано при розгалуженні дозволяє зменшити кількість даних, що необхідно передавати такій програмі.[7]

Для задач, щопотребуютьінтернет-з'єднання для надсилання результатів виконання також потрібна черга із пріоритетами, щоб відправити в першу чергу результати виконання критично важливих задач. Також потрібно в таких системах застосовувати механізми стисненняданих.

Деякі задачі можуть виконуватися автономно і потребують інтернет-з'єднання лише у випадку, коли виникає деяка помилка або інша нестандартна ситуація. В цьому випадку система використовує зворотній зв'язок для прийняття рішення щодо подальших дій. Чим більше нестандартних ситуацій та винятків ми зможемо передбачити, тим менше імовірність, що нам знадобиться корекція. Крім виключень та помилок система може використати зворотній зв'язок для запиту додаткових даних, тобто на етапі виконання задачі може знадобитися додаткова інформація. Очевидним вирішенням даної проблеми є надсилання всієї подібної інформації на етапі ініціалізації задачі (якщо це можливо). Це збільшує обсяг інтернет-трафіку, а також потребує додаткової пам'яті для зберігання даних, які можуть і не знадобитися. В цьому випадку надійність (і відповідно - робота в автономному режимі) забезпечуються надлишковістю, що призводить до додаткових витрати ресурсів системи.

В умовах нестабільного живлення нашої інформаційної системи, виконання багатьох операцій потрібно здійснювати в режимі транзакцій. Відключення живлення означатиме, що транзакцію необхідно відмінити. Тобто за рахунок транзакцій ми гарантуємо, що в нас не буде частково виконаних завдань. Для впровадження механізму транзакцій нам також знадобиться журнал. [8]

В залежності від прикладних задач, які вирішує інформаційна система, користувачам також рекомендується завантажити для локального зберігання словники іноземних мов, карти місцевості, компас, годинник, секундомір,календар, сканер штрих кодів та QR-кодів, довідники та підручники, що містять всі необхідні для виконання прикладних задач формули та інструкції, середовища розробки/компілятори, утиліти для перегляду файлів різних форматів (pdf, powerpoint, пакету microsoftoffice), утиліти для шифрування, розшифрування повідомлень та утиліти для цифрового підпису, телефонний довідник, а також програму для установки додатків із APKфайлів, так як при обмеженій пам'яті мобільного пристрою ми можемо зберігати частину додатків в пам'яті мобільного пристрою, а частину на зовнішньому носії, і переносити їх з зовнішнього носія на мобільний та в інший бік відповідно до поточних завдань, які необхідно виконувати системі.

6. Модель

Очевидно, що при наявності вибору між забезпеченням енергонезалежності системи та засобами підтримки функціонування системи в умовах відключень електроенергії, потрібно обрати саме перший варіант. Якщо сторона захисту може забезпечити функціонування системи в режимі 24/7 за рахунок паливних генераторів чи акумуляторних батарей, які вдається повністю зарядити протягом періоду наявності електроенергії, то матеріали, викладені в даній статті вам не знадобляться.

Для вибору найкращого проекту використовують метод Фішберна або Уея. Обидва методи є лінійними, мають однакову обчислювальну складність і розраховані не невелику кількість показників ефективності (зазвичай не більше 5), при цьому показники утворюють строго спадну послідовність, тобто нема двох показників з однаковою вагою. Значення коефіцієнтів для обох методів **не залежать** від предметної області, а визначаються виключно за формулою. Їх

значення залежить лише від кількості показників. так для 5 показників ефективності ці коефіцієнти матимуть наступний вигляд:

$$\frac{1}{15};\frac{2}{15};\frac{3}{15};\frac{4}{15};\frac{5}{15} - \textit{для методу Фішберна}$$

$$\frac{1}{25};\frac{3}{25};\frac{5}{25};\frac{7}{25};\frac{9}{25} - \textit{для методу Уея}$$

Ці коефіцієнти є послідовністю звичайних дробів, чисельники яких утворюють арифметичну послідовність, а знаменник містить їх суму. Достатньо використовувати один із методів, другий використовують лише тоді, коли за першим отримують дуже близькі оцінки. Якщо показників більше 5, їх групують. Використовувати методи Фішберна та Уея для проектів з більшою ніж 5 кількістю показників недоцільно так як послідовність коефіцієнтів дуже швидко спадає і показники, які будуть знаходитися в кінці цієї послідовності практично не впливатимуть на результат, так як відповідні коефіцієнти будуть дуже близькі до 0.

Бальний метод, при якому коефіцієнти обирають з урахуванням специфіки предметної області (а значить можуть бути 2 або більше однакових коефіцієнта), застосовувати складніше, так як при ньому виникають проекти з однаковими оцінками (за рахунок однакових коефіцієнтів). Так, при наявності однієї пари однакових коефіцієнтів і однієї трійки однакових коефіцієнтів, значення яких відрізняються від зазначеної пари, кількість проектів становитиме

$$Q = 2! \cdot 3! = 12 \text{ проектів} \tag{6.1}$$

Якщо ж електроенергія відсутня протягом тривалих періодів часу внаслідок стихійних лих чи війни, а можливості придбати генератори чи акумуляторні батареї немає через брак ресурсів або їх дефіцит на ринку (джерела живлення потрібні не лише нам), тоді потрібно використовувати приведену тут модель.

Почнемо із розрахунку запасу енергії. Все обладнання нашої системи розділимо на безпосередньо обчислювальне та мережеве. Система може виконувати деякі задачі навіть без підключення до мережі, тому постійне живлення мережевого обладнання не знадобиться.

$$E = P_s t_s + P_n t_n \tag{6.2}$$

де E – загальний запас енергії

P_s та P_n – потужність (швидкість споживання енергії на одиницю часу для s-системи, n – мережевого обладнання), t_s та t_n - час функціонування системи.

при цьому $t_s \neq 0$, $t_s \geq t_n$

перше обмеження пояснюється тим, що нам нема сенсу підтримувати мережеве обладнання, якщо система не працює

Друге обмеження є наслідком того, що мережеве обладнання використовується разом із основним.

Тепер розглянемо проблему функціонування системи в умовах, коли через відсутність електроенергії вона не отримує нові завдання/команди.

У випадку простою (система виконала всі заплановані задачі, а нові отримати не може) величина збитків розраховується за формулою $L = BL - EP$,

де BL (businesslosses) – втрати пов'язані із невиконанням системою прикладних задач, а EP-(energypreserved) – вартість зекономленої електроенергії, так як на період простою систему можна тимчасово відключити повністю звівши до нуля всі витрати або скоротивши їх до рівня, коли подібними витратами можна знехтувати. Очевидно, що при $L < 0$ відключення системи для збереження енергії є задовільним рішенням, так як будь-які витрати пов'язані із бізнес процесами будуть меншими ніж вартість підтримки системи в робочому стані. Подібну ситуацію ми спостерігаємо зазвичай в системах, які існують в неприбуткових організаціях, тобто в системах, які фінансуються ззовні (дотаційні). В більшості випадків $L > 0$,

але потрібно пам'ятати, що дана формула дійсна для одноразового інциденту. Якщо відключення енергії повторюються (неважливо – регулярно чи ні), то зекономлена енергія EP додається до загальної енергії, що буде доступна під час наступного відключення живлення,

тобто ми можемо конвертувати зекономлену сьогодні енергію в функціонування системи завтра при заданій потужності протягом деякого проміжку часу. І протягом даного часу система приноситиме прибуток, тобто ми виграємо не просто вартість електроенергії, а додану вартість від продукту, який система виробляє, споживаючи цю енергію.

Якщо з економічних причин варіант із простоєм системи нас не влаштовує, то ми можемо в якості альтернативи скористатися режимом функціонування із надлишковим виконанням завдань. В цьому режимі після того, як закінчилися всі надіслані задачі, система повинна почтаки виконувати нові задачі без вказівки ззовні (від користувачів), а при поверненні електроенергії деяка частка цих задач виявиться “потрібною”, тобто сюди увійдуть задачі, які були б поставлені перед системою її користувачами, якби живлення не було відключено. В цьому випадку система продовжує споживати електроенергію від генераторів/акумуляторних батарей, але з меншим коефіцієнтом корисної дії. Сторона захисту може побудувати граф можливих сценаріїв поданих задач. Це зважений орієнтований граф, де кожне ребро (дуга) відповідатиме за імовірність надходження відповідної задачі (вершина, в яку входить дуга графу).

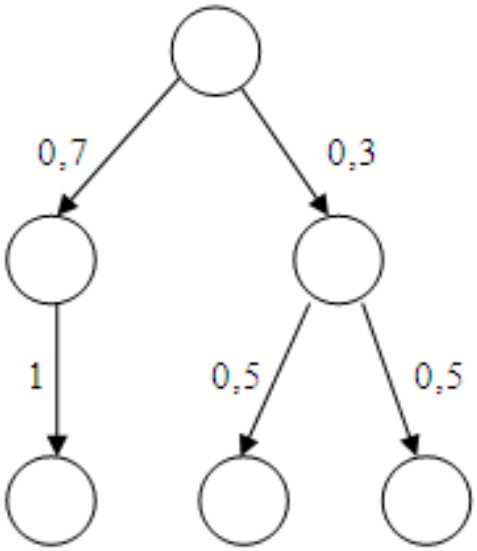

Рис.2 – Граф поданих задач

у випадку, коли предтсавлений граф є деревом, розрахувати імовірності буде просто. Достатньо перемножити всі числа на дугах, що ведуть до заданої вершини. Таким чином ми маємо імовірнсоті 0,7, 0,15 та 0,15, що в сумі дорівнює 1 і утворює повну групу подій.

В цьому прикладі стороні захисту доцільно припустити, що надійде задача, яка стосується лівої гілки графу з імовірністю 70%. В правій гілці дві рівноімовірні задачі по 15%. Але це лише імовірнсоті. Для обчислення прибутку чи збитку потрібно враховувати тривалість виконання кожної із задач та прибуток від них. А для того, щоб вся наша стратегія із надлишковістю працювала, потрбіно щоб виконана під час простою задача покривала витрати на функціонування системи, і не лише на власну електроенергію, а і на електроенергію, що була витрачена на всі задачі, які ми не вгадади. Якщо після відновлення живлення виявиться, що жодна із виконаних задач не була насправді потрібною, то ви втрачаємо ще більше ніж при простої системи, бо тепер ми не отримуємо прибуток від обслуговування бізнес процесу і ми також при цьому витратили електроенергію, яку не зможемо використати завтра для виконання нових задач.

Виглядає це наступним чином

$$Pr = \sum_{i=1}^{n}(\alpha_i BP_i - EP_i) \tag{6.2}$$

де Pr (profit) – загальний прибуток,
BP_i (businessprofit) прибуток від виконання і-тої задачі,

EP_i - вартість енергії, що була витрачена на дану задачу,

$$\alpha_i = \begin{cases} 1, якщо задача вгадана правильно \\ 0, в протилежному випадку \end{cases} \quad (6.3)$$

Відповідно до даної формули у порушника є два реалістичних і один гіпотетичний спосіб спричинити збитки для нашої системи саме в контексті обмежених енергоресурсів. (всі стандартні кібератаки залишаються можливими і продовжують становити загрозу)

по-перше порушник може намагатися вплинути на α_i, тобто зменшити кількість задач, які ми правильно вгадали. Наприклад порушник може тимчасово стати звичайним користувачем системи і протягом деякого часу подавати в систему однотипні задачі, оплачуючи їх виконання. Після деякого проміжку часу, коли ми вважатимемо імовірність надходження задач від даного користувача близькою до 100%, а природу задач передбачуваною в достатній мірі (наприклад обробка відео файлів із одного плейлиста на youtube), ми починаємо самостійно назначати ці задачі системі ще до того, як ми отримаємо запит від самого користувача. В день проведення атаки користувач просто не подає свій пакет задач і ми витратили час та енергію марно. Дана атака стає ще більш загрозливою, коли її виконує група порушників, так як в такому випадку їм буде легше вичерпати всі значення α_i

По-друге порушник може впливати на BP_i. проведення атаки відбувається за схожим сценарієм, але тепер замість того, щоб просто зникнути (як користувач) в день проведення атаки, він продовжує видавати себе за користувача, але відмовляється платити за виконану задачу, тому що вона “виконана неякісно”. Продовжуючи тему із обробкою відео із деякого плейлиста, порушник може розташувати відеофайли в плейлисті таким чином, щоб в день проведення атаки, коли він збирається відмовитися від оплати, опрацьовані відео містили захищену законом про авторське право музику або це можуть бути відео в іншому форматі, обробка якого нашою системою не передбачена. В будь-якому випадку сторона захисту може недоотримати прибуток, а в найгіршому випадку втратити кошти у зв’язку із виплатами штрафів та репутаційними збитками.

Третій гіпотетичний випадок – це вплив на EP_i, тобто на кількість енергії, що необхідна для виконання задачі. Даний випадок є гіпотетичним, тому що тут порушнику практино завжди потрібно вийти за межі моделі “порушника інформаційної безпеки” і використовувати зовнішні по відношенню до системи засоби впливу. Для того, щоб різко збільшити вартість енергії, порушнику потрібно влпинути на вартість енергоносіїв, тобто палива, яке ми використовуватимемо в генераторах в періоди відключення системи. Для того, щоб вплинути на тарифи, за якими ми купуємо енергоресурси, порушнику потрібно мати досить значний вплив хоча б на рівні населеного пункту, де розташована система. Або може знадобитися терракт для знищення складу, на якому зберігаються енергоносії, що призведе до їх дефіциту і відповідного стрибка в ціні. В контексті даної статті розглядати подібні сценарії недоцільно.

На α_i та BP_i можна впливати і не вдаючи із себе користувача системи. Замість того, щоб самостійно подавати системі задачі (і оплачувати їх) порушник може просто провести атаку на відмову в обслуговуванні на законних користувачів системи. В день проведення атаки ми просто не надішлють свої задачі до кінця робочого дня, і якщо інформація швидко застаріває, то наступного дня виконана робота виявиться марною. Парадоксально, але відповідно до описаної вище формули атаку можна провести **припинивши атаку**! Якщо протягом тривалого часу порушник не підпускав до системи деяку підмножину користувачів, це може легко вплинути на наші уявлення про вагу на дугах графу. В день проведення атаки порушник просто припиняє атаку на відмову в обслуговуванні на користувачів, і виявляється, що заплановані нами задачі становлять значно меншу частку того, що потрібно законним користувачам системи. Продовжуючи тему із обробкою відео, припустимо що наша система надає послуги двох видів – 1) вилучення аудіо із відео, щоб залишилося відео без звуку і 2) вилучення відеоряду із відео, щоб залишилася лише аудіодоріжка. Обмежуючи доступ користувачів, яким потрібно вилучити аудіо, порушник змінює імовірності на дугах в графі задач. Припинивши атаку він створить ситуацію, при якій аномально великій частці користувачів сайту знадобляться відеофайли без звуку, а наша система їх не підготувала. Ну а вартість задачі може змінитися, коли зростає попит на альтернативу. Розглядаючи наш приклад із обробкою відео, система може надавати послуги із додаванням до відео водяного знаку та послуги із цифрового підпису. обидві послуги

дозволяють захистити відеофайл від несанкціонованого використання та поширення, але при використанні водяного знаку ми змінюємо кожен кадр відео, а при додаванні цифрового підпису – впливаємо лише на метадані.

Висновок

Використання інформаційних систем в умовах обмеження живлення створює додаткові ризики, пов'язані із якісною та кількісною нестачею механізмів захисту. Деякі із цих ризиків можна мінімізувати, використовуючи надлишковість, спланувавши роботу системи в автономному режимі. При плануванні потрібно враховувати тип проблеми із живленням (обмежене за часом, чи постійне але з перебоями), вплив проблеми живлення на мережеве обладнання (автономна робота та робота із інтернет-з'єднанням), тип задач (важкі задачі потребують більше енергії), їх послідовність за взаємозалежність. Використання мобільних пристроїв як тимчасової альтернативи ПК є можливим, але потребує попередньої підготовки даного пристрою і у будь-якому випадку зробить даний пристрій більш вразливим до атак. Перспективним напрямками для подальших досліджень є підвищення гетерогенності систем для повної або часткової відмови від технологій віртуалізації (віртуальні машини створюють значен навантаження на процесор та пам'ять) та використання хмарних техголгій (сегмент розміщений в хмарі може мати гарантоване джерело живлення, а локальний сегмент працювати з обмеженнями енергоресурсів).

REFERENCES

1. M. McCune, *The complete solutions guide for every Linux/Windows system administrator!* Prentice Hall PTR, 2000, 416p
   http://armstrong.craig.free.fr/eBooks/Prentice%20Hall/Prentice%20Hall%20Integrating%20Linux%20and%20Windows.pdf
2. H. Kaminaga "Improving linux startup time using software resume (andother techniques)" *Linux symposium* .Vol2/2006 pp25-34.
   https://www.kernel.org/doc/mirror/ols2006v2.pdf#page=25
3. W Dargi "A Stochastic Model for Estimating the Power Consumption of a Processor" *IEEE TransactionsonComputers*, vol. 64, no. 5/2015, pp. 1311-1322,
   https://ieeexplore.ieee.org/abstract/document/6783802
4. K Schüller, K Staňková, F Thuijsman "GameTheoryofPollution: National Policies and Their International Effects". *Games*. vol 8(3)/2017: p30.
   https://www.mdpi.com/2073-4336/8/3/30
5. T McGill, N Thompson "Old risks, new challenges: exploring differences in security between home computer and mobile device use" *Behaviour&InformationTechnology* vol36(11)/2017, pp. 1111–1124
   https://www.tandfonline.com/doi/abs/10.1080/0144929X.2017.1352028
6. Q Burke *et al*., "Enforcing Multilevel Security Policies in Unstable Networks" *IEEE Transactions on Network and Service Management*, vol. 19, no. 3/2022, pp. 2349-2365
   https://ieeexplore.ieee.org/abstract/document/9779867
7. C Young, M D Smith "Improvingtheaccuracyofstaticbranchpredictionusingbranchcorrelation"*ACM SIGOPS Oper. Syst. Rev.*, 28, 5/1994, pp 232–241.
   https://dl.acm.org/doi/abs/10.1145/381792.195549
8. A Kolli, S Pelley, A Saidi, P M Chen, T F Wenisch "High-Performance Transactions for Persistent Memories" *SIGARCH Comput. Archit. News*, vol 44, 2/2016, pp399–411.
   https://dl.acm.org/doi/abs/10.1145/2872362.2872381

KireienkoOleksandr *PhD student*
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute",37 Peremohy Ave., Kyiv-56, Ukraine, 03056

Information protection in the conditions of limited energy resources

Relevance: The protection of information and the system where such information is processed and stored is a complicated task even under optimal conditions. Adding any constraints leads to the appearance of new problems. The constraint on energy resources forces us to change an approach to information protection, abandon certain solutions and change the way the way we use information system. The primary focus of this article is on information systems that use networking hardware. Networking hardware is viewed as part of the system and requires power supply. The problem of limited energy resources is examined at two levels – restricted to a single system and globally when such systems compete for limited resources. The violator is aware of the system's functioning under constrained energy resources but in general case doesn't kbnow how dire the situation is.
Goal:to design a model of inforamtion oritection with aforementioned constraints, to select task with high ration of reward for completion to energy usage, to create a reserve of energy for further use when the system is operating again, to estimate the risk of overloading the grid.
Research methods:vgraph theory, game theory, Fishburn's and Way's methods for selection of best project.
The results: the model was designed and 2 strategies of energy resources distribution were found
Conclusions: The problem of lack of energy resources in general case is solved by the means of introduction of redundamcy in computational and memory resources. The conduction of an attack and the halting of one both will affect the distribution of tasks that are queued to the syste, by legitimate users. Affecting the cost of energy resources is possible but is outside of the scope of this article.

Keywords*: energy resources, autonomous power source, networking hardware, mobile devices, task scheduling, data backup*

Надійшла у першій редакції 17.09.2024, в останній - 21.10.2024.
The first version has been received on 17.09.2024, the final version – on 21.10.2024.