ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ КІБЕРБЕЗПЕКИ ТА ЗАХИСТУ ІНФОРМАЦІЇ

СИСТЕМ ТА ТЕХНОЛОГІЙ КІБЕРБЕЗПЕКИ

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Технологія захисту інформаційних систем від атак на ланцюг поставок»

зі спеціальності *125 Кібербезпека та захист інформації*
(код, найменування спеціальності)

освітньо-професійної програми *Інформаційна та кібернетична безпека*
(назва програми)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ Євген ЩИБУН
(підпис)

Виконав: здобувач вищої освіти групи БСДМ-61
Щибун Євген
(прізвище, ім'я)

Керівник к.т.н., доцент ВЛАСЕНКР Вадим
(науковий ступінь, вчене звання, прізвище, ім'я)

Рецензент _____
(науковий ступінь, вчене звання, прізвище, ім'я)

Київ 2024

ЗМІСТ

Стор.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

AST – Application security testing

DAST – Dynamic Application Security Testing

DoD – Department of Defense

EMD – Engineering and Manufacturing Development

HW – Hardware

ICT – Information and Communications Technology

KP – Key Practice

MSA – Materiel Solution Analysis

O&S – Operations and Support

P&D – Production and Deployment

RASP – Runtime Application Self-Protection

SCM – Supply Chain Management

SCRM – Supply Chain Risk Management

SE – Systems Engineering

SW – Software

TARA – Threat Assessment and Remediation Analysis

TD – Technology Development

PPP – Program Protection Plan

ВСТУП

Supply Chain Management (SCM) допомагає виробничим підприємствам управляти потоком товарів та послуг, охоплюючи всі потрібні аспекти: логістику, розробку та виробництво продукції, а також рішення та системи SCM.По суті, ланцюжок поставок – це процес, спрямований на надання клієнтам товарів чи послуг, а управління ланцюжком постачання – це координація такого процесу. Проста версія ланцюжка поставок включає компанію, її постачальників та клієнтів.

Атака на ланцюг поставок – це кібератака, що здійснюється, використовуючи довіру між компанією-виробником та її клієнтами. [8-13] Вона завдає шкоду організації, орієнтуючись на слабкі місця у ланцюгу постачання. Зловмисник ставить під загрозу один або кілька компонентів процесу розробки чи доставки. Така атака може використовувати підроблені чіпи, що задіяні у виробництві комп'ютера або мережевого обладнання. Або, атака може запровадити скомпрометований код у програмному засобі, що не викликає підозри. Є цілий ряд різних сценаріїв та методів, але головне, що ця атака використовує надійні канали для проникнення, щоб досягти своїх цілей. Тому тема магістерської роботи в умовах сьогодення є своєчасною та актуальною.

Об'єкт дослідження –захист інформаційної системи організації від атак на ланцюг поставок.

Предмет дослідження – технологія захисту інформаційної системи організацій від атак на ланцюг поставок.

Метою роботи – роботи є підвищення рівня захищеності інформаційної системи організації шляхом розробки й впровадження технології захисту від атак на ланцюг поставок.

Наукове завдання:

- дослідити аналіз сутності ланцюга поставок в інформаційній системі організації;

- визначити вплив атак на ланцюжок поставок в інформаційній системі організації;
- дослідити технологію створення каталогу шаблонів атак, оцінки загроз та оцінки кіберзагроз щодо вибору ефективних контрзаходів в процесі життевого циклу системи SCRM при використанні ланцюга поставок в інформаційній системі організації.
- Дослідити можливості технології Imperva RASP щодо захисту від атак на ланцюг поставок.

Методи дослідження: опрацювання літератури за даною темою, аналіз експлуатаційної документації, міжнародних стандартів та їх порівняння, практичне використання засобів захисту від supply-chain-attack.

Практичне значення одержаних результатів: запропоновано порядок застосування технології захисту від атак на ланцюг поставок інформаційної системи організації із застосуванням технології Imperva RASP, а також розроблено рекомендації фахівцям з кібербезпеки щодо застосування даної технології.

1 АНАЛІЗ ПРОБЛЕМИ ЗАХИСТУ ІНФОРМАЦІЙНОЇ СИСТЕМИ ОРГАНІЗАЦІЇ ВІД АТАК НА ЛАНЦЮГ ПОСТАВОК

1.1. Аналіз сутності ланцюга поставок в організаціях

Базовий рівень управління ланцюгом поставок (supply chain) - це всі дії, які необхідно зробити, щоб покупець отримав свій товар. Обробка покупки та замовлення, виробництво, перевезення, управління запасами - і багато інших процесів при наданні послуг. Все починається з аналізу ринку: необхідно дізнатися, які продукти потрібні клієнтам, коли та в якій кількості. Далі необхідно послідовно пройти всі етапи від вибору постачальника до виробництва та логістики. Це комплексний та складний механізм. Для того щоб це все працювала максимально ефективно, кожного партнера (ланка в ланцюжку) необхідно вбудувати в єдину гнучку систему (рис.1.1) [1, 14].

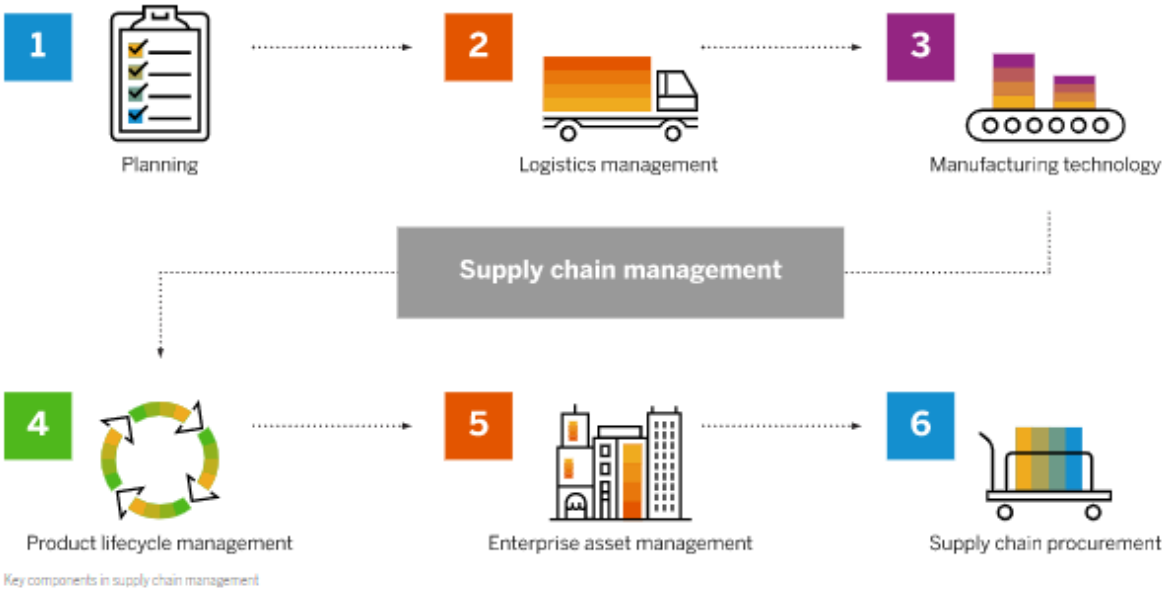

Рис 1.1. Етапи управління ланцюгом поставок

Починаючи з 2022 року весь світ усвідомив критичну важливість ланцюжків постачання. Недоставлені товари, неможливість знайти необхідну сировину для виробництва — ці та подібні ситуації продемонстрували і компаніям, і споживачам,

що ланцюжки мають бути стійкими та масштабованими. Зараз компанії по всьому світу аналізують свої глобальні ланцюжки постачання та технології для управління ними. Головне питання – що можна зробити, щоб гарантувати успішність бізнесу у майбутньому.

Оптимізований підхід до SCM (Supply Chain Management) здатний трансформувати бізнес. Завдяки мінімізації відходів та надлишків при одночасному зниженні витрат та підвищенні ефективності компанії можуть стати більш конкурентоспроможними. Якісне та персоналізоване обслуговування – перевірений спосіб підвищити лояльність клієнтів. А автоматизація процесів зробить їх швидшими, інтелектуальнішими і продуктивнішими.[1]

Глобальні ланцюжки поставок, збудовані за останні кілька десятків років, дозволяли користуватися дешевшою сировиною та нижчою заробітною платою в деяких країнах, отримуючи бізнес-перевагу. Але часи змінюються. Правила торгівлі, тарифи – все перебудовується, часто непередбачувано. Політичне майбутнє невизначене. Клієнти хочуть мати повну інформацію про походження та екологічність продуктів; це стосується і сировини, і упаковки, і навіть палива, що використовується в автомобілі, що доставляє замовлення до дверей. На щастя, технології не відстають. У сучасних ланцюжках поставок для оптимізації виробництва використовуються великі дані, що генеруються кожною ланкою ланцюжка. Більше того, кожна ланка може розробити власне рішення для управління ланцюжком постачання.

Розглянемо детально компоненти управління ланцюжком поставок [1], яке показано на рис.1.1.

Планування ланцюжка поставок

Плануванням ланцюжка поставок називається процес прогнозування потреби у продукті та координація всіх ланок ланцюжка для відповідного постачання. Крім прогнозування та планування попиту, сюди включені планування поставок, планування потреб у матеріалах, планування виробництва, а також планування збуту та операційної діяльності. У сучасному SCM ці процеси зазвичай об'єднані в єдине інтегроване бізнес-планування.

Управління логістикою

Управління логістикою - це транспортування та зберігання товарів на всіх етапах ланцюжка поставок, починаючи з постачання сировини та виробництва та закінчуючи доставкою готової продукції до магазинів або клієнтів. Воно може охоплювати навіть етапи обслуговування, повернення та утилізації товарів. До його бізнес-функцій відносяться управління транспортуванням як вхідним, так і вихідним управлінням транспортним парком, управління складами, контроль запасів та обслуговування клієнтів.

Технологія виробництва

Що стосується виробництва в рамках SCM, одним з основних завдань є забезпечення максимальної економічності за збереження (і підвищення) якості, екологічності та задоволеності клієнтів. Системи на основі штучного інтелекту (ШІ) та Інтернету речей (IoT) у ланцюжку поставок дозволяють збирати та аналізувати великі дані для оптимізації та автоматизації виробничих процесів.

Управління життєвим циклом продуктів (PLM)

Управління життєвим циклом продукту (PLM) - це процес управління продуктом на всіх етапах, від формування ідеї та проектування до виробництва, обслуговування та утилізації (або переробки). Програмні системи PLM об'єднують усі ці процеси, координуючи роботу різних підрозділів у межах всього підприємства, та формують основний масив інформації про кожен продукт протягом його життєвого циклу.

Управління активами підприємства (EAM)

Управління активами підприємства - це процес управління та обслуговування фізичних активів по всьому ланцюжку поставок, від заводської робототехніки до транспортних засобів, що використовуються для доставки. Для трансформації EAM активно використовуються датчики Інтернету речей, міжмашинна взаємодія (M2M) та цифрові двійники. Ефективність, працездатність та безпека активів зростають, а профілактичне та діагностичне обслуговування стає простіше. Деякі підключені активи можуть прогнозувати несправності та самостійно виконувати

технічне обслуговування - аж до вибору постачальника та замовлення деталей, необхідних для продовження їх життєвого циклу.

Закупівлі в рамках ланцюжка постачання

Закупівлі – це процес придбання матеріалів, товарів та послуг потрібної якості для задоволення потреб бізнесу за обговореною справедливою ціною. Головною проблемою для відділів закупівель та вибору постачальників є прогнозування точних обсягів замовлень: як дефіцит, так і надлишки можуть зашкодити бізнесу. Системи SCM із технологією машинного навчання та прогнозної аналітики дозволяють точніше визначити необхідний обсяг закупівель.

Розглянемо переваги управління ланцюжком постачання:

Зростання продуктивності. Системи управління активами підприємства та діагностичне обслуговування підвищують ефективність обладнання та систем. Компанія може усунути вузькі місця, покращити робочі процеси та збільшити продуктивність. Крім того, автоматизовані процеси та гнучкий аналіз даних прискорюють відвантаження та скорочують час постачання.

Скорочення витрат на ланцюжок постачання. Прогнозна аналітика позбавляє необхідності гадати в умовах, коли кожна помилка може коштувати дуже дорого — жодних зайвих витрат на занадто великі запаси та ризики дефіциту. Технології Інтернету речей спрощують управління активами, допомагаючи вибудувати ефективні робочі процеси кожної ситуації. Точність прогнозування також зростає: зменшується кількість автомобілів з неповним завантаженням та нескоординованих маршрутів доставки, керування транспортним парком стає ефективнішим.

Підвищення гнучкості та стійкості ланцюжка поставок. Тенденції та ситуація на ринку можуть змінитися будь-якої миті, тому велике значення мають стійкі системи SCM, здатні адаптуватися до будь-якої ситуації. Обробка даних у реальному часі та інтелектуальний аналіз допомагають менеджерам ланцюжка поставок перерозподіляти обладнання та персонал між робочими процесами. У будь-який момент можна зібрати зворотний зв'язок від клієнтів та негайно вжити

заходів. Віртуальні запаси та інтелектуальні складські процеси допомагають збалансувати попит та пропозицію.

Підвищення якості продукції. Відділи досліджень та розробок отримують прямий доступ до відгуків покупців, що дозволяє проектувати продукти відповідно до потреб клієнтів. Команди досліджень, розробки та виробництва можуть спиратися на дані машинного навчання та аналітики, покращуючи дизайн продукту та швидко реагуючи на поточні тенденції та побажання клієнтів.

Зростання якості обслуговування клієнтів. Найкращі SCM-практики - гнучкі, адаптивні, орієнтовані на клієнта. Сучасна система SCM пропонує конкурентні переваги, дозволяючи компаніям враховувати зворотний зв'язок від клієнтів та тенденції на всіх етапах, від проектування та виробництва до логістики останньої милі, постачання та повернення.

Система управління ланцюгами постачання (SCM-система) — прикладне програмне забезпечення, призначене для автоматизації та управління всіма етапами постачання підприємства та для контролю за весь рух товарів: закупівля сировини та матеріалів, виробництво, розповсюдження продукції.[2]

SCM-системи можуть виконуватись у різних варіантах:

- як самостійні, тиражовані SCM-системи;
- як рішення, реалізовані у вигляді складових частин ERP-систем;
- як унікальні системи, що створюються для конкретних підприємств.

ERP (Enterprise Resource Planning, планування ресурсів підприємства) — організаційна стратегія інтеграції виробництва та операцій, управління трудовими ресурсами, фінансового менеджменту та управління активами, орієнтована на безперервне балансування та оптимізацію ресурсів підприємства за допомогою спеціалізованого інтегрованого пакета прикладного програмного забезпечення, що забезпечує загальну модель даних і процесів для всіх сфер діяльності.

ERP-система – конкретний програмний пакет, який реалізує стратегію ERP.

Стратегія ERP охоплює майже всі напрямки діяльності підприємства, але ERP-система все ж таки не замінює CRM-систему, яка контролює зовнішні відносини, і PLM-систему, що управляє інтелектуальною власністю.

Характерні особливості ERP-стратегії:

Використання єдиної транзакційної системи для переважної більшості операцій та бізнес-процесів підприємства. Всі операції зводяться в єдину базу для подальшої обробки та отримання в реальному масштабі часу збалансованих планів.

Тиражованість: забезпечення можливості застосування одного і того ж програмного пакета для різних організацій (можливо, з різними налаштуваннями та розширеннями).

Підтримка в єдиній системі безлічі валют та мов.

Підтримка кількох юридичних осіб, кількох підприємств, кількох облікових політик, різних схем оподаткування в єдиній системі; це необхідно для її застосування у корпораціях, у т. ч. транснаціональних.

Програмний пакет, який реалізує стратегію ERP, зазвичай виконується у вигляді набору модулів, перелік яких може варіюватися в залежності від розміру та особливостей підприємств, на яких впроваджується ERP-система. Стандартний набір виглядає так:

Фінанси та бухгалтерський облік.

Управління персоналом.

Склад.

Продаж.

Управління взаємовідносинами з клієнтами (CRM).

Закупівлі.

Управління ланцюгом поставок (SCM).

Виробництво.

1.2. Аналіз ризиків організацій від атак supply chain

Атаки на ланцюжок поставок можуть завдати шкоди організаціям, окремим відділам або цілим галузям, вибираючи та атакуючи небезпечні елементи ланцюжка поставок програмного забезпечення [2, 3].

Ланцюжок поставок програмного забезпечення складається з:

Елементів процесу життєвого циклу розробки програмного забезпечення (SDLC), включаючи системи складання, середовища розробки та тестування.

Програмного забезпечення з відкритим вихідним кодом або інше програмне забезпечення, яке використовується як компонент корпоративного програмного забезпечення.

Платформи з відкритим вихідним кодом, які використовуються безпосередньо підприємствами, такі як WordPress або Magento.

Постачальники, які надають професійні послуги, консультації чи послуги з розвитку

Партнери, які зберігають чи обробляють дані від імені підприємства

Хмарні послуги (включаючи IaaS, PaaS та SaaS)

Попередні постачальники підприємства, які все ще володіють даними компанії або мають доступ до IT-систем.

Більшість організацій мають обмежену видимість свого ланцюжка постачання програмного забезпечення. Будь-яка третя сторона, яка недостатньо захищена та надає програмне забезпечення або послуги великим організаціям, є ризиком атаки на ланцюжок поставок.

Найчастіше зловмисники шукають найслабші ланки в ланцюжку поставок — наприклад, вони націлені на дрібних постачальників без засобів контролю за кібербезпекою або компонентів з відкритим вихідним кодом з невеликою спільнотою або слабкими заходами безпеки.

Більшість атак на ланцюжок поставок викликано додаванням бекдорів у законне та сертифіковане програмне забезпечення або компрометування систем, що використовуються сторонніми постачальниками. Ці атаки важко виявити за допомогою існуючих засобів захисту від кібербезпеки.

Розглянемо основні загрози щодо атак на ланцюжок постачання (supply chain attack).

Кібератака на ланцюжок поставок компанії – це не лише короткострокова проблема, але вона має також і середньо- та далекострокові наслідки:

1. Втрата зовнішньої інформації. Цей результат очевидний: кібер-злочинці отримують у свої руки інформацію, що належить користувачам платформи, яка має бути безпечним середовищем.

2. Втрата внутрішньої інформації. Це не лише проблема для клієнтів чи користувачів. Це також проблема для компанії як такої, оскільки її корпоративна інформаційна безпека серйозно постраждає, і вона може постраждати від крадіжки внутрішніх даних або конфіденційної інформації, яка є життєво важливою для повсякденної діяльності компанії.

3. Репутація. Якщо у користувача вкрали його дані із зовнішньої платформи, то логічно припустити, що навряд чи він надалі довірятиме цій платформі. І це одна із найсерйозніших проблем, з якою зіткнулися British Airways та Ticketmaster: як тепер повернути довіру клієнтів?

4. Санкції. З 25 травня 2018 року всі організації відповідальні за дотримання вимог нового європейського законодавства щодо захисту персональних даних GDPR, та відповідні органи уважно стежать за компаніями, які порушують ці вимоги. Одним із наслідків крадіжки персональних даних може стати визнання того, що компанія порушила законодавство. У цьому випадку будь-яка компанія може зіткнутися з багатомільйонними (у євро!) штрафами.

Розвивається тенденція

Як раніше зазначалося у звітах провідних компаній з кібербезпеки за останні роки цілком ймовірно, що ми побачимо набагато більше випадків атак на ланцюжки поставок (supply chain attack), враховуючи їх ефективність, їх вплив (вони можуть швидко поширюватися на мільйони систем), а також довіра користувачів, які вважають скомпрометоване ПЗ або платформи законними та безпечними.

Для боротьби з цією проблемою необхідно здійснювати в реальному часі моніторинг всіх процесів у системі, але моніторинг повинен фокусуватися на впровадженні всього ПЗ у компанії, забезпечуючи цілісність та конфіденційність ланцюжка поставок та захищати інформаційну систему організації [4, 5, 6].

Найчастіше найбільша небезпека від кібер-атак може лежати в самому центрі IT-систем компанії, а тому вона може впливати не тільки на інформаційну безпеку самої компанії, але також на безпеку та конфіденційність інформації та даних третіх осіб, користувачів, клієнтів та ін. , схоже, що ця тенденція активно розвивається, тому компанії повинні уважно стежити за тим, щоб у їхньому ланцюжку постачання все працювало «як годинник».

Технології які повинні забезпечувати захист інформаційної системи організації повинні покривати атаки OWASP топ-10 і більше.

Для цього можна застосовати технологію RASP, яка може захистити від складних загроз і нульових днів, які включені, але не обмежуючись прикладами, які наведені:

- Командна ін'єкція
- Clickjacking
- Міжсайтові сценарії (XSS)
- Підробка міжсайтових запитів (CSRF/XSRF)
- Порушення доступу до бази даних (розширений SQLI)
- Ін'єкція HTML
- Підробка методу HTTP
- Поділ відповіді HTTP
- Небезпечні файли cookie
- Небезпечний транспорт
- JSON Injection
- Великі запити
- Реєстрація конфіденційної інформації
- Неправильно сформовані типи вмісту
- OGNL Injection
- Обхід шляху
- SQL Injection
- Реєстрація конфіденційної інформації
- Небезпечний транспортний протокол

• Несанкціонована мережева діяльність
• Неперехоплені винятки
• Неперевірені запити
• Вразливі залежності
• Слабка аутентифікація
• Слабке керування кеш-пам'яттю браузера
• Слабка криптографія та шифри
• XML External Entity Injection (XXE)
• XML Injection

1.3. Аналіз структури та підходів до вирішення проблеми атак на ланцюг поставок

Структура та підходи спрямовані на SCRM в системі придбання і, зокрема, на тему атак на ланцюжок поставок. Мета полягає в тому, щоб розробити розуміння моделей атак, які проводяться для використання вразливостей у ланцюжку поставок в інформаційній системі організації та протягом життєвого циклу розробки системи.

Завдання фремворку полягає у необхідності зібрати широкий спектр інформації про атаку на ланцюг поставок та структурувати її в корисну структуру для задоволення потреб різноманітної спільноти SCRM.

Все це необхідно робити для того, щоб надати уявлення про атаки на ланцюг поставок щодо зловмисної вставки протягом повного життєвого циклу придбання, якого на сьогоднішній день не було виявлено. Такий підхід структурує та кодифікує атаки ланцюга поставок, використовуючи шаблони атак, які включають пов’язану інформацію про загрози та вразливості.

Очікується, що отриманий каталог моделей атак:

• допоможе інформаційній системі організації придбати та підтримувати системи, які є менш вразливими до атак ланцюга поставок, усуваючи зловмисне введення в ланцюг поставок.

• надавати інформацію для зосередження аналізу загроз ланцюга поставок та оцінювати вразливості, які виконують інженери програмного забезпечення.

Розглянемо, що є основою для розробки фреймворка та збору даних про атаку. Спочатку було включено широкий спектр дослідницьких джерел, щоб проаналізувати проблемний простір з точки зору SE, який включав атаки зловмисної вставки через ланцюжок поставок, мережеві атаки на інформаційні системи, зв'язок між вразливими місцями ланцюга поставок, які дозволяють зловмисним вставкам та вразливостям, імплантованими зловмисниками, які дозволяють здійснювати атаки під час операцій, а також потенційні пом'якшення та компроміс між ризиками та витратами, необхідними для вибору контрзаходів для ефективного зменшення ризику безпеки [5, 6].

У той час як вищезгадані заходи зосереджені в основному на захисті від системних атак, інші джерела контрзаходів включали захист від зловмисної вставки через ланцюг поставок. Найбільш помітним серед них є Посібник із ключових практик SCRM (DoD-SCRM 2010), який описує 32 ключові практики (КР) як пом'якшення ризику для загроз ланцюга поставок. Існують інші, більш безпосередньо пов'язані й постійні зусилля інженерів та спеціалістів із безпеки в рамках кількох піддисциплін системної безпеки. Ці зусилля спрямовані на вирішення загроз, вразливостей та атак на різних рівнях [7]. Наприклад:

• Національний інститут стандартів і технологій (NIST.) Посібник описує події загрози, спрямовані на інформаційні системи, і надає компіляцію репрезентативних прикладів подій зі сторони зловмисника [8].

• Міністерство внутрішньої безпеки спонсорує постійні зусилля щодо розширення та підтримки загальнодоступного каталогу, який надає загальні перерахування та класифікацію шаблонів атак (CAPEC) типових методів використання програмного забезпечення (MITRE Corporation 2012). Шаблони атак CAPEC фіксують і передають точку зору зловмисника ПЗ, похідну від концепції

шаблонів проектування, застосовуваних у деструктивному, а не конструктивному контексті та створених на основі поглибленого аналізу реальних операцій ПЗ [10, 11].

• Корпорація MITRE розробила методологію оцінки загроз та аналізу відновлення (TARA) для виявлення та оцінки кіберзагроз та загроз ланцюга поставок та вибору ефективних контрзаходів (Wynn et al. 2011). Методологія TARA спирається на каталог змагальних тактик, методів і процедур (TTP), який був побудований в основному на основі взаємодії з програмами інформаційної системи.

Ключові практики SCRM разом із загальним, наскрізним відображенням системи ланцюга поставок були використані, щоб забезпечити широкий набір моделей атак на ланцюг поставок. Кожена з 32 ключових поставок відстежує принаймні одну атаку. І існує щонайменше 8 моделей атак, визначених для кожної з точок атаки в карті ланцюга поставок.

Посібник із ключових практик SCRM та інші джерела, згадані вище, також були використані для складання початкового набору контрзаходів як підтвердження концепції загального процесу відстеження атак у ланцюжку поставок до дійових рекомендацій щодо зниження ризику [9,12]. Контрзаходи, які були визначені:

• безпечне керування конфігурацією програмного забезпечення;

• запобігання або виявлення несанкціонованого доступу до критичних компонентів;

• мови програмування, орієнтовані на безпеку;

• стандарти та огляди дизайну та кодування, орієнтованих на безпеку;

• red teaming ланцюга постачання;

• надійна доставка;

• надійні механізми доставки;

• теги відстеження та мітки безпеки;

• каталог, встановлений по всьому ланцюжку поставок

• масовий запас запчастин

• кількість постачальників;

• надійні постачальники;

• анонімність того хто придбав;

• електромагнітний / тепловий аналіз;

• обмеження мережевого трафіку;

• візуальний огляд;

• криптографія;

• наглядність ланцюга постачання;

• довіра персоналу;

• безпека оновлення програмного забезпечення.

Таким чином, як висновок зазначимо про необхідність створення каталогу конкретних шаблонів атак, оцінки загроз та аналізу відновлення для виявлення та оцінки кіберзагроз та загроз ланцюга поставок та вибору ефективних контрзаходів в процесі життевого циклу системи SCRM.

2 МЕТОДИ ТА ЗАСОБИ ЗАБЕЗПЕЧЕННЯ ЗАХИСТУ ІНФОРМАЦІЙНОЇ СИСТЕМИ ОРГАНІЗАЦІЇ ВІД АТАК НА ЛАНЦЮГ ПОСТАВОК

2.1. Схеми атак на ланцюг поставок supply chain

Приклади атак на ланцюжок поставок включають вставку зловмисного програмного забезпечення в бібліотеки з відкритим кодом і заміну підроблених компонентів HW у відділі-отримувачі на нижньому рівні ланцюга поставок. Перший використовує процес придбання, щоб створити вразливість дизайну (пов’язаний з відкритим вихідним кодом), а другий використовує слабкість процесу приймального відділу. Беручи до уваги такі широкомасштабні проблеми, корисно розглянути, що саме буде визначено як сферу дії цих зусиль, а що визначено як те, що поза її межами [12, 13].

У результаті був створений каталог конкретних шаблонів атак на ланцюг поставок, розмежованих наступним чином із наступними областями:

Об'єкт атак, що розглядаються:

Придбані або підтримувані компоненти інформаційно-комунікаційних технологій (ІКТ) або системи ІКТ.

Розглядаються наступні типи атак на ланцюг поставок:

Навмисна руйнівна установка (в тому числі заміна, зміна та впровадження шкідливого ПО) HW, SW або FW в критично важливі компоненти ІКТ.

Розглянуті часові рамки атак включають:

У будь-який час протягом життєвого циклу придбання системи, включаючи попереднє придбання, придбання або підтримку.

Точками атаки в ланцюжку поставок є:

Розташування (див. рис. 2.1): місця розробки системи та програмного забезпечення і їхні внутрішні процеси та середовища; наприклад, інтегровані середовища розробки (IDE).

— шкідлива діяльність, яка відбувається в будь-якому місці ланцюга поставок, включаючи інструменти розробки та процеси, що належать/використовуються цим сайтом/об’єктом.

— Розташування ланцюга постачання включає офіс програми, головного підрядника та всі рівні субпідрядників/субпостачальників та інтеграторів (включені до цих категорій діяльність з підтримки на місцях; наприклад, склади запчастин та діяльність з підтримки програмного забезпечення; та їх постачальники).

— Між місцями (див. рис. 2.2): зв'язки ланцюга поставок.

— Шкідлива активність, яка відбувається у фізичному потоці між місцями ланцюга поставок (тобто логістичні мережі покупців і постачальників).

— Шкідлива діяльність, яка відбувається в рамках потоку інформації та даних ланцюга поставок (тобто зовнішні ІКТ/IDE-середовища покупця та постачальника).

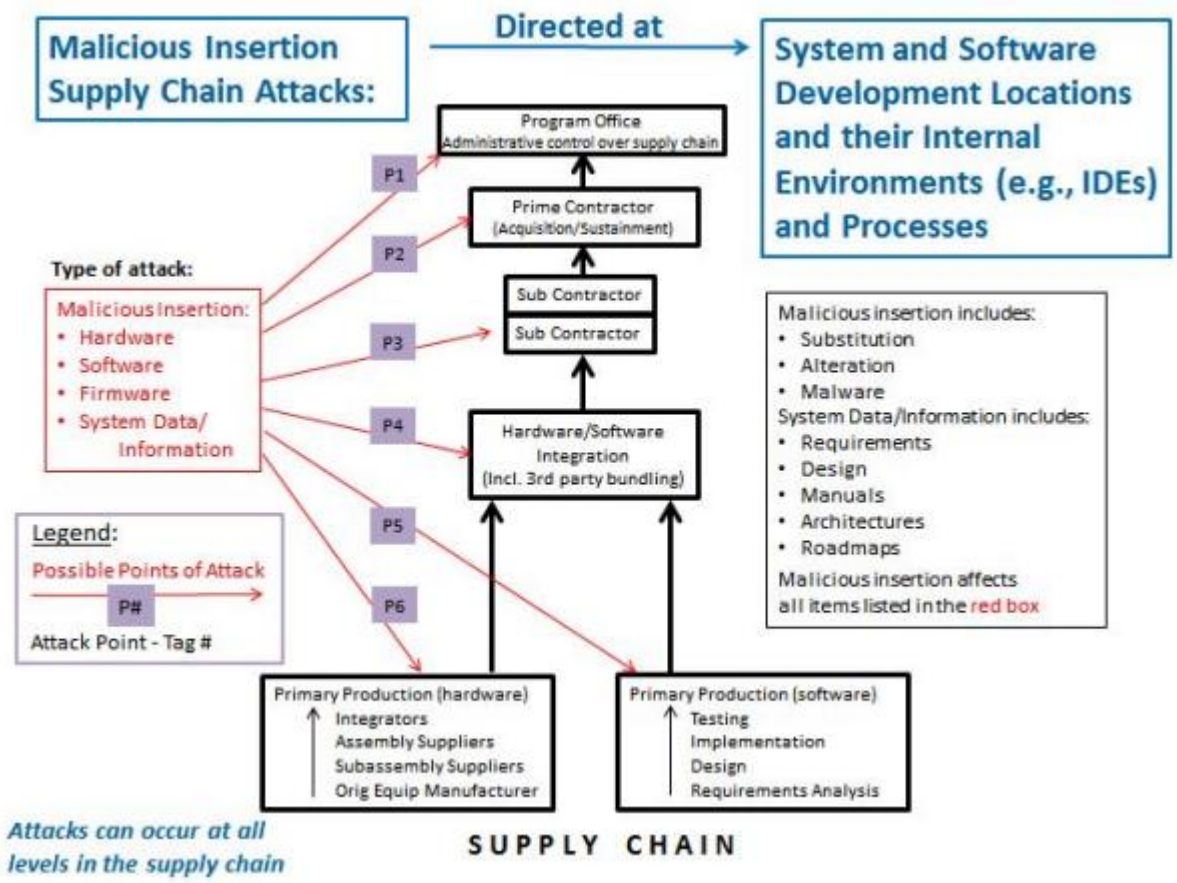

Рис.2.1. Точки атаки – розташування ланцюга поставок

З огляду на такий обсяг, метою було зібрати, структурувати та розробити моделі атак, які використовуються для зловмисної вставки в критичні компоненти протягом усього життєвого циклу системного придбання, шляхом виявлення недоліків, які можна використовувати в ланцюжку поставок інформаційної системи, за допомогою загальної, наскрізної системи ланцюга поставок, як показано на рис. 2.1 і рис.2.2.

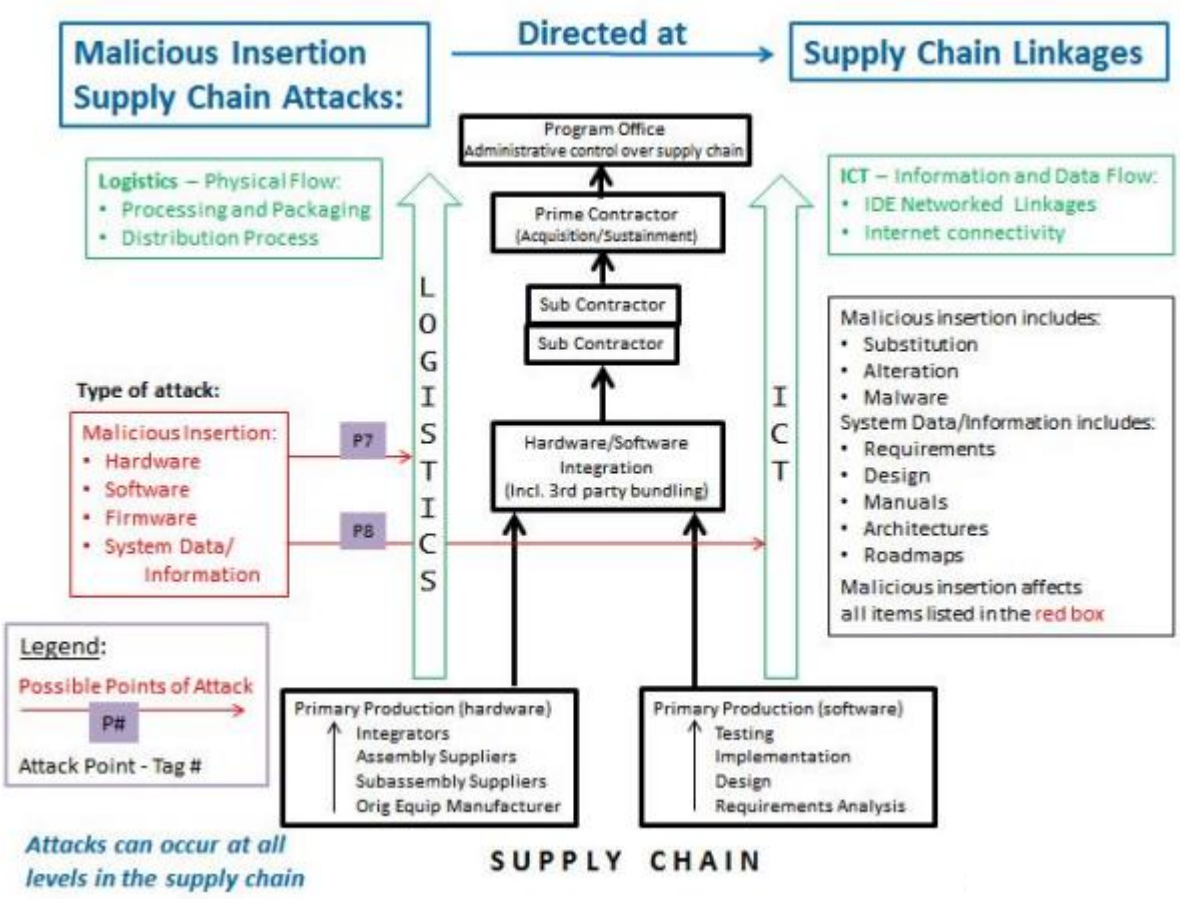

Рис 2.2 Точки атаки – зв’язки ланцюга постачання.

Наступні приклади атак забезпечують додаткову ясність щодо обсягу:

Ланцюг постачання проти системи:

За обсягом: атаки ланцюга постачання на систему, яка підтримується/надається.

Приклад: імплантація бекдора в програмне забезпечення системи під час розробки або обслуговування.

Поза межами дії: мережеві, інсайдерські або фізичні атаки на систему під час операцій.

Приклад: використання бэкдора в системі SW, який був імплантований під час розробки або обслуговування.

Системи підтримки:

В Рамках: атаки ланцюга постачання на системи підтримки "першого порядку" (безпосередньо пов'язані з розробкою системи) для придбання.

Приклади: зловмисно змінені компілятори; зловмисне програмне забезпечення, вставлене в середовище розробки HW; зловмисно змінені інструменти програмування, що програмуються в польових умовах (FPGA).

Поза межами: атаки ланцюга постачання на системи підтримки «другого порядку» для придбання.

Приклад: зловмисне вставлення коду в систему доставки та одержання, щоб порушити процеси розповсюдження.

Приклади поза межами сфери:

• Шкідливе вилучення в ланцюжку постачання, включаючи втрату:

передових технологій;

інтелектуальної власності;

невизначеної контрольованої технічної інформації.

• Розгляд загроз безпеці та вразливостей, які не ґрунтуються на атаках.

Приклад 1: Існуючі недоліки в системі (наприклад, ненавмисні вразливості програмного забезпечення), які потенційно можуть бути пом'якшені контрзаходами ланцюга поставок.

Приклад 2. Використання підрядником постачальника для інтегральної схеми з критичною функцією (ASIC), що відрізняється від відомого/надійного постачальника, який раніше був зазначений у плані закупівель підрядника.

Деталі каталогу шаблонів атак. Каталог шаблонів атак був створений з використанням різних джерел даних та інформації ланцюга поставок, та на основі TTPs TARA, елементів ланцюга поставок CAPEC та загроз змагань, складених NIST [3]. Отримана в результаті загальна сукупність атак доводить все, що вже було зафіксовано в TARA і CAPEC для ланцюга поставок, до витонченого рівня деталей. Дані NIST були видобуті та переведені на предмет їх актуальності та застосовності для отримання системою DoD.

Змагальні атаки складаються з багатьох атрибутів, включаючи джерело змагальної загрози, метод, який використовує супротивник, дію, що викликає зловмисне вставлення, і мету противника. Ці зусилля розробили структуру атаки на ланцюжок поставок для структурування та опису моделей атаки на ланцюжку поставок, де кожен зразок розробляється контекстними даними – наданими у вигляді 12 специфічних атрибутів, які структурують та кодують шаблон атаки. Каталог включає зміст для 41 моделі атак, які можна аналізувати різними способами для підтримки аналізу загроз та оцінки вразливості.

12 атрибутів атаки, які формують кожен із 41 шаблонів атаки:

• ID атаки (унікальний ідентифікаційний номер)

• Точка атаки (розташування ланцюга постачання або зв'язок)

• Цільова фаза (фаза життєвого циклу придбання)

• Тип атаки (зловмисне введення технічного, програмного забезпечення або системної інформації/даних)

• Акт нападу ("що")

• Вектор атаки ("як")

• Походження атаки (“хто”)

• Ціль атаки ("чому")

• Вплив атаки (наслідок у разі успіху)

• Посилання (джерела інформації)

• Загроза (змагальна подія, спрямована на ланцюг поставок)

• Вразливі місця (слабкі сторони, які можна використовувати)

Короткий опис кожного атрибута наведено в дужках вище. Детальний опис наведено на рис. 2.3. Позначення тегів точки атаки («P#»), наведені на рис. 2.3, графічно проілюстровані на рис. 2.1 і 2.2.

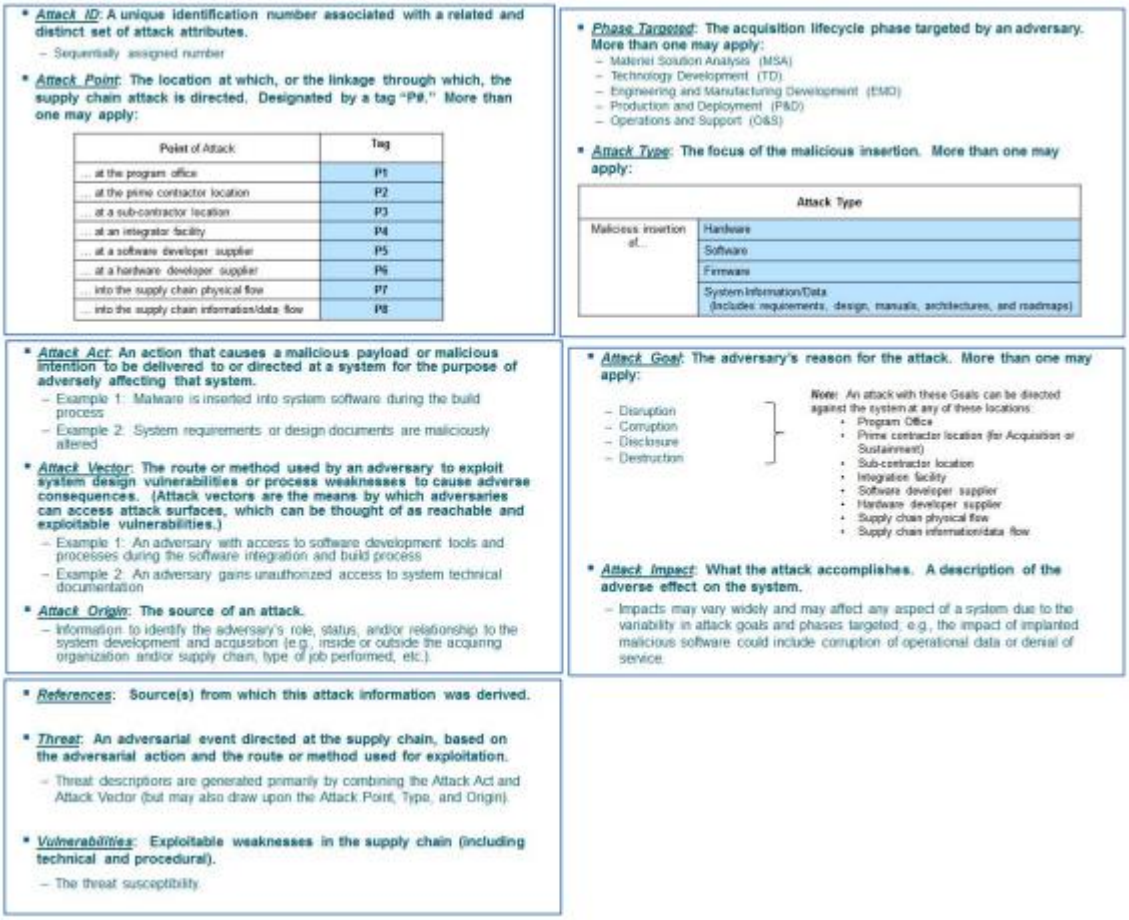

Рис.2.3. Визначені атрибути атаки

2.2. Побудова шаблону атаки

Шаблони атаки були побудовані шляхом заповнення атрибутів атаки у межах інформації про атаки, зібраною та структурованою з різних джерел. Часто було корисно побудувати графічне зображення ключових атрибутів атаки під час її розробки. Наприклад, рис. 2.4 ілюструє атаку зловмисного введення ПЗ в будь-яке з інженерних середовищ розробників/підрядників даного ПЗ під час будь-якої фази життєвого циклу після Milestone-B. (Це атака A3 в каталозі.)[5]

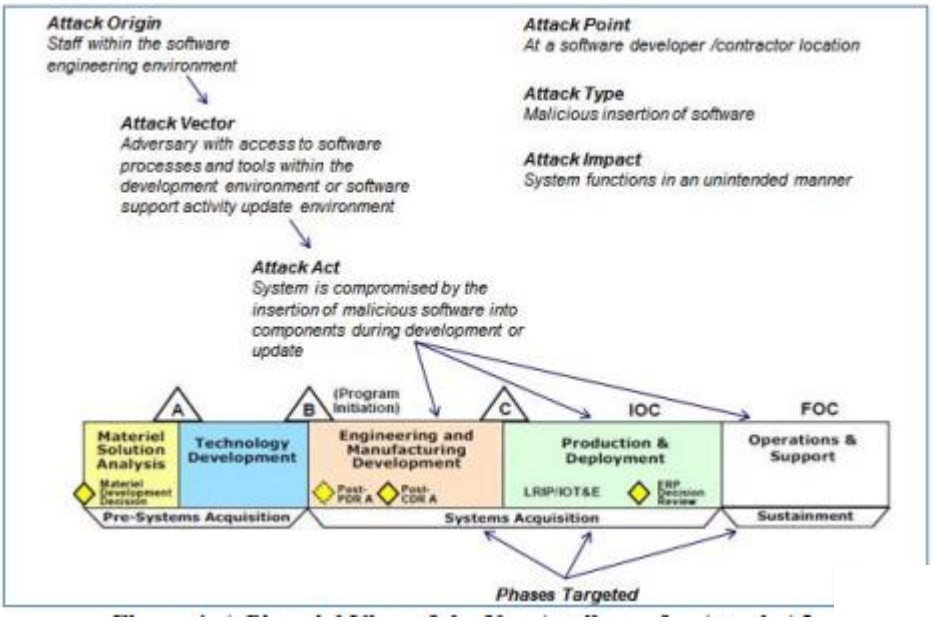

Рис.2.4. Ілюстративний вигляд ключових атрибутів для атаки А3

Ключова інформація про атаку, проілюстрована на рис. 2.4, - це що, як і хто. Закон про атаки повідомляє вам, на який тип зловмисної вставки націлена атака. Вектор атаки описує, частину “як” – маршрут або метод, який використовує супротивник. Походження атаки - це частина “хто” - статус, роль або відношення супротивника до програми.

На основі цієї інформації було розроблено решту атрибутів для атаки А3, а на рис.2.5 наведено знімок атаки з остаточного каталогу.

Attack Identifier: A3
Target (Attack Type): Hardware: Firmware: Software: Yes Sys Information or Data:
Description (Attack Act): System is compromised by the insertion of malicious software into components during development or update.
Attack Vector: Adversary with access to software processes and tools within the development environment or software support activity update environment.
Attack Origin: Staff within the software engineering environment.
Attack Goal: Disruption: Yes Disclosure: Yes Corruption: Yes Destruction:
Attack Impact: System may function in a manner that is unintended.
References: Based on NIST SP 800-30; page E-4
Threat: An adversary with access to software processes and tools within the development or software support environment can insert malicious software into components during development or update/maintenance.
Vulnerabilities: The development environment or software support activity environment is susceptible To an adversary inserting malicious software into components during development or update.
Attack Points: Program Office: Software Developer: Yes Prime Contractor: Yes Hardware Developer: Sub-Contractor: Yes Physical Flow: Integrator Facility: Yes Information Flow:
Applicable Life Cycle Phases: Materiel Solution Analysis: Technology Development: Engineering and Manufacturing Development: Yes Production and Deployment: Yes Operations and Support: Yes

Рис.2.5. Шаблон атаки А3 [5]

2.3. Способи аналізу каталогу атак

Структуру та зміст каталогу можна проаналізувати різними способами, щоб отримати уявлення про поточні атаки ланцюга поставок. Існують різні способи, за допомогою яких каталог підтримуватиме аналіз та оцінку атак на ланцюжок поставок. Наприклад, на рис. 2.6 показано 12 розподілів 41 шаблонів атак як за типами критичних компонентів, які потребують захисту, так і за фазами життєвого циклу, на які атаки спрямовані. Існує досить рівномірний розподіл між HW та SW; також важливими є шкідлива вставка в програмне забезпечення і системна інформація/дані [9,11,14].

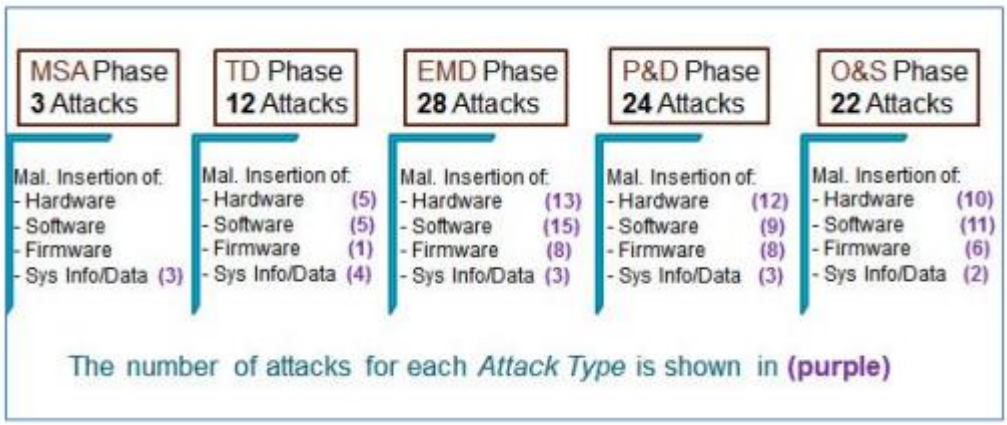

Рис. 2.6. Аналіз типів атак за фазами

Рисунок 2.6. показує аналіз типів атак за фазами. Хоча не дивно, що фаза EMD чутлива до найбільшої кількості атак (див. рис. 2.6), дуже цікаво вивчити, як деякі з цих атак застосовні протягом усього життєвого циклу. На рис 2.7 показано, що багато атак застосовні на кількох етапах.

Хоча не дивно, що фаза EMD чутлива до найбільшої кількості атак (див. рис. 2.6), дуже цікаво вивчити, як деякі з цих атак застосовні протягом усього життєвого циклу. На рис. 2.7 показано, що багато атак застосовні на багатьох етапах.

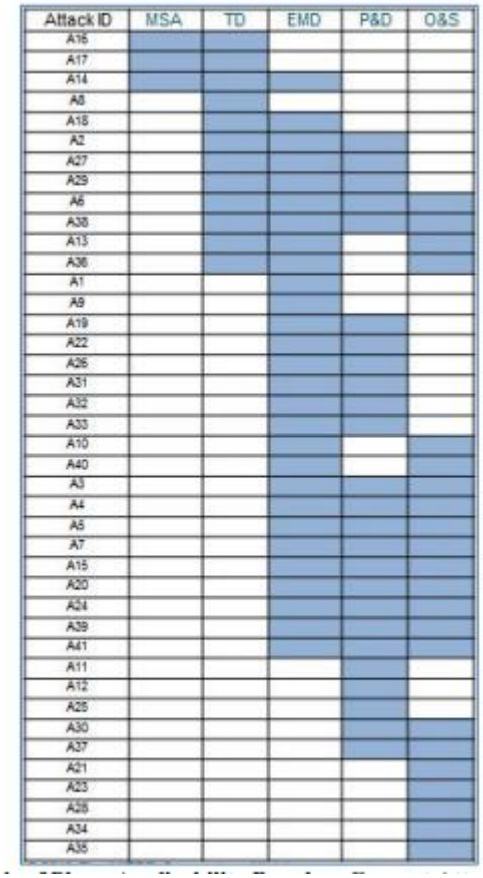

Рис.2.7. Аналіз застосовності фази на основі розуміння поточної атаки

З цього аналізу можна зробити кілька висновків:

• Більшість атак застосовуються на кількох етапах.

• Існує значна кількість атак фази TD.

о Планування цих атак має відбуватися під час фази MSA.

• Раннє планування пом'якшення наслідків має бути спрямоване на використання економічно ефективного захисту протягом усього життєвого циклу.

• Понад 2/3 атак застосовні до фази EMD.

• Більшість атак, що застосовуються до P&D, також застосовуються на попередніх етапах.

• Існують важливі атаки, спрямовані лише на ланцюг поставок.

Аналіз, показаний на рис. 2.8, демонструє, що можна дізнатися про потенційні точки атаки для кожної атаки.

Рис.2.8. Аналіз застосовності точки атаки

З даного аналізу зрозуміло, що:

• Близько половини атак можуть відбуватися або в офісі програми, або в місцях головного підрядника.

• Більшість атак, що застосовуються до головних, також застосовні до нижчих рівнів.

• Більшість атак, що стосуються субпідрядників, також застосовні до об'єктів інтеграторів.

• Постачальники розробників програмного забезпечення та постачальники розробників програмного обладнання піддаються однаковій кількості атак.

В цьому підрозділі наведено кілька основних аналізів, шаблони атак можна відфільтрувати та структурувати в інші уявлення, щоб підтримувати специфічний для програми розгляд конкретних типів атак ланцюга поставок.

2.4. Концепція використання інструменту підтримки прийняття рішень

Каталог шаблонів атак забезпечує розробку зловмисної вставки HW, SW, FW, а також системної інформації та даних у критичні компоненти системи DoD, яка отримує або підтримується. Програми придбання можуть знайти цей збірник корисним для:

• Оцінки та встановлення рівня захисту програми та ресурсів SSE (System Security Engineering).

• Керівництво аналізом TSN (Trusted Systems and Networks).

• Вибір та перевірка контрзаходів.

• Допоміжний аналіз випадків зловживання.

• Виконання тестування на проникнення в ланцюг поставок, щоб перевірити, наскільки ланцюг поставок насправді захищений від зловмисного введення.

Даний підрозділ зосереджено на потенційному прикладному підході для підтримки інженерів програми придбання, коли вони виконують аналіз TSN. Як інструмент підтримки прийняття рішень, вміст структури можна аналізувати та застосовувати різними способами, щоб зосередитися на конкретних типах атак на ланцюжок поставок і інформувати, з технічної та процедурної точки зору, про аналіз загроз ланцюга поставок та оцінку вразливості в повному обсязі життєвого циклу.

Як приклад сценарію для ілюстрації того, як може використовуватися структуру, припустимо, що ваші критично важливі компоненти системи були ідентифіковані за допомогою аналізу критичності, і ви хочете використовувати каталог для визначення потенційних атак зловмисної вставки. У інформаційній системі організації є багато критично важливих компонентів програмного забезпечення, тому необхідно зосереджені на потенційних атаках на програмне забезпечення, які можуть статися під час фази EMD і далі. Рисунок 2. 9 фільтрує та сортує всі моделі атак відповідно до типів критичних компонентів та фаз, на які націлені атаки.

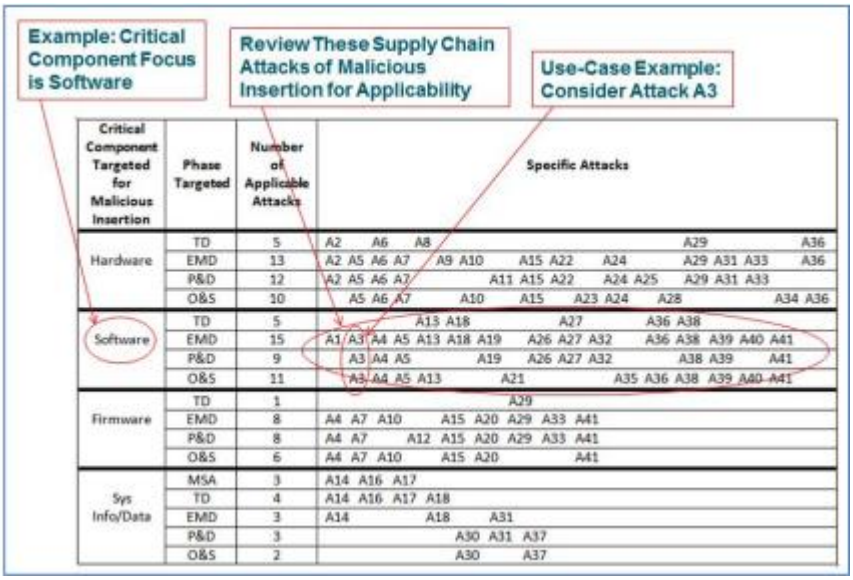

Рис.2.9. Сценарій використання атак для розгляду

Для цього варіанту використання необхідно переглянути всі атаки, які обведені великим червоним овалом на рис. 2.9 (тобто A1, A3, A4, A5, A13 тощо), щоб отримати цілісне уявлення про потенційні атаки зловмисної вставки, націленої на програмне забезпечення. Більшість атак застосовні протягом кількох фаз життєвого циклу. Боротьба з такими атаками на початковому етапі може обмежити витрати на захист ланцюга поставок пізніше. Деякі атаки застосовні під час фази TD, і, хоча ви безпосередньо зацікавлені в фазі EMD і далі, може виявитися корисним розглянути, що могло бути зроблено для захисту під час фази TD і чи цей тип атаки все ще представляє загрозу для вашої програми.

Вибравши атаку A3 для продовження цього прикладу використання, можливо потім дослідитие ключові атрибути цього шаблону атаки (які були графічно представлені на рис. 2.4).

На основі цього аналізу необхідно визначити застосовність атаки A3 до структури ланцюга постачання вашої програми та вашого середовища розробки програмного забезпечення з урахуванням того, як вони будуть змінюватися з часом на етапах EMD, P&D та O&S фаз придбання.

Кожен зразок атаки в каталозі містить конкретну інформацію про загрози та вразливості, пов’язані з цією атакою. На малюнку 2.5 представлено знімок атаки A3 з каталогу. Вивчивши атрибути загрози та вразливості для атаки A3, можна

легко побачити, що акт нападу атаки та вектор атаки (з допоміжною інформацією з джерела атаки) є перш за все тим, що входить до опису загрози та вразливості, які представляє атака A3.

Для атаки A3 така інформація з каталогу може виявитися корисною для аналізу TSN та для подальшої розробки плану захисту програми (PPP) (DASD SE 2011):

• Загроза: зловмисник, який має доступ до програмних процесів і інструментів у середовищі розробки або підтримки програмного забезпечення, може вставити шкідливе програмне забезпечення в компоненти під час розробки або оновлення/обслуговування.

• Вразливості. Середовище розробки або середовище діяльності з підтримки програмного забезпечення є чутливим до того, що зловмисник може вставити шкідливе програмне забезпечення в компоненти під час розробки або оновлення.

Висновок: передбачувані види використання та переваги структури атак на ланцюг поставок і каталогу включають наступне:

• Користувачі можуть зосередитися на конкретних типах атак на ланцюжок поставок, які можуть завдати шкоди їхнім системам, чи то під час придбання, так і в умовах середовища.

• Дані моделі атак можуть бути відсортовані за будь-яким атрибутом, який користувач вважає релевантним (наприклад, тип атаки, цільова фаза або точка атаки).

• Серед користувачів – спеціальні програми (та їх підрядників), яким доручено виконувати аналіз TSN для захисту критичних компонентів.

2.5. Створення шляхів захисту від атак на ланцюг поставок

При використанні в різних програмах або доменах структура атак на ланцюг поставок та каталог можуть покращити узгодженість і одноманітність аналізів і звітів, пов’язаних із SCRM. Цей каталог інформації може стати основою майбутньої характеристики атак на ланцюг поставок.

Наступні потенційні кроки можуть включати:

• Програмне забезпечення.

o Перегляньте приклади використання та/або підтримайте розробку випадків зловживання за допомогою вибраних програм.

o Використовуйте завдання для надання інформації про концепції впровадження та покращення структури та її змісту (наприклад, для інформування про аналіз TSN та покращення PPP (Program Protection Plan).

• Партнерства

o Створюйте партнерства, що включають інтереси Microelectronics і Software Assurance, щоб забезпечити якнайширше охоплення атак на ланцюжок поставок.

o Прослідкуйте, щоб атака ланцюга поставок і робота з контрзаходів відповідали наскрізним потребам різноманітних складових.

• Стратегія перехідного періоду

o Визначте, як цю роботу можна структурувати та інституціалізувати, щоб максимізувати зручність використання та отримати користь.

o Вивчіть альтернативні підходи та намагайтеся включити цю роботу до існуючих/розробних каталогів та інструкцій.

Таким чином, для атак на ланцюжок поставок згідно викладеного необхідно застосовувати технології, які за допомогою аналізу дозволять отримувати контроль над поведінкою між компонентами програми на ланцюг поставок .

Атаки на ланцюжок поставок вислизають від традиційного їх виявлення.

Вбудовуючи невідомий або спеціально написаний шкідливий код у довірені програми, зловмисникам потрібні додаткові інвестиції, але їм вдається оминати продукти периметра та WAF.

Команді безпеки не вистачає контексту поведінки програми

Традиційні рішення забезпечують видимість лише вхідного та вихідного трафіку додатків. Атаки на ланцюжок поставок продовжуватимуть залишатися непоміченими доти, доки поведінка програми не буде вивчена.

Поведінка сучасної програми складна і у міру того, як нові технології поширюються в корпоративних середовищах, поєднання мікросервісів, API-інтерфейсів та контейнерів ставить нові завдання перед фахівцями з безпеки.

Тож надалі дослідимо технологію захисту від атак на ланцюжок поставок за допомогою Imperva.

Захист програм під час виконання Imperva (RASP) використовує полегшений модуль безпеки для аналізу активності всередині програми та блокування небажаних дій, таких як сторонні бібліотеки, що встановлюють мережне підключення до зовнішнього сайту для управління та контролю (C&C).

Imperva RASP захищає програми, середовище виконання, сервери, залежність з відкритим вихідним кодом та сторонні бібліотеки. Він розгортається за лічені хвилини, легко підключаючись до програми, не вимагаючи будь-яких змін у коді та не вимагаючи постійних оновлень сигнатур та інше.

RASP є ідеальним в місцях, де програма повинна працювати в автономному середовищі без будь-якої мережі або доступу в Інтернет, але з локалізованим доступом, який, як і раніше, потребує певної форми перевірки та безпеки. RASP забезпечить принаймні захист програми від шкідливих запитів та мінімальну пропускну здатність або використання ресурсів у разі застарілих програм.

3 ТЕХНОЛОГІЯ ЗАХИСТУ ДОДАТКІВ ПІД ЧАС ВИКОНАННЯ В ІНФОРМАЦІЙНІЙ СИСТЕМІ ВІД АТАК НА ЛАНЦЮГ ПОСТАВОК

3.1 Технологія самозахисту додатків під час виконання (RASP)

Безпека інформаційної системи та програм останнім часом набула власної форми, і досі не існує єдиного стандартного визначення їх безпеки. В цьому розділі буде досліджено технологію та різні елементи безпеки додатків у інформаційній системі організації, а також запропоновано для захисту від атак на ланцюг поставок інновацію, яка зосереджена на захисті програм: самозахист додатків під час виконання (RASP) [5,6].

Розглянемо структуру архітектури підприємства

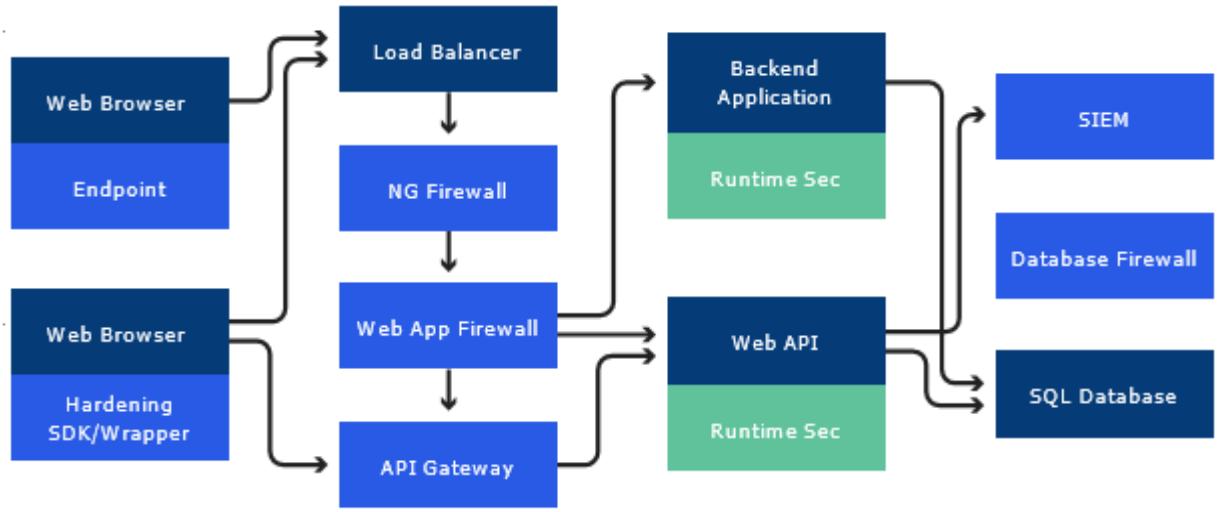

Рис. 3.1. Структура архітектури підприємства, яке використовує веб-додатки як свій бізнес

Більшість підприємств сьогодні мають як застаріле, так і нове програмне забезпечення, яке керує їхнім бізнесом, і ці програми знаходяться в складному середовищі, що охоплює мережу, додаток, базу даних та операційну систему. Завдяки додаткам які є локальні, так і ті що в хмарі, організації перебувають у різних станах цифрової трансформації. Чим старше підприємство, тим більш буде фрагментоване його середовище. Фрагментація відбувається в середовищі

розробника або DevOps з його кількома мовами програмування (JAVA, .NET, Node.js тощо) і кількома базами даних. Фрагментація також спостерігається в безпеці додатків з декількома і часто розрізненими рівнями контролю безпеки, починаючи від статичного, динамічного та інтерактивного тестування безпеки (SAST, DAST, IAST) до самозахисту додатків під час виконання (RASP), а також захист на основі периметра за допомогою мережевих брандмауерів і брандмауерів веб-додатків (WAF).

Фрагментовані організації та їх контроль

Успішна програма безпеки додатків вимагає кількох доменів і ресурсів для взаємодії та обміну інформацією. Окрім технічних компонентів (додатків, інструментів інфраструктури, засобів контролю безпеки), задіяні дві великі організації: розробники (Builders) та рахисники (Defenders).

Builders. Команди зазвичай знаходяться в підрозділах бізнесу, технологій і продуктів. Їхня мета — якомога швидше запустити програми у виробництво під тиском прискорення циклів випуску. Середовище, в якому вони працюють, зазнає трьох серйозних порушень:

1. IT-зміни. Результатом є зростання програмно-визначених центрів обробки даних, віртуалізація, контейнеризація, публічні та приватні хмари, SaaS, IaaS і PaaS у новому наборі завдань, які відволікають розробників від головної мети: швидкої розробки додатків.

2. DevOps. Більшість команд розробників додатків зараз об’єднують зусилля зі своєю IT/CIO-організацією, щоб перейти до гнучкого середовища DevOps, допомагаючи швидкістю, гнучкістю розробки та зворотним зв’язком.

3. DevSecOps. Розробники часто змушені тісно співпрацювати з Defenders (командами організації CSO / CISO) або адміністраторами безпеки, щоб виправити або пом’якшити відставання вразливостей. У той час як такі процеси, як Secure SDLC, інструменти сканування, такі як SAST/DAST/IAST, і тестування на проникнення вказують на проблеми всередині додатків, тиск часу, бюджету, продуктивності та проблеми виправлення часто змушують розробників запускати код у виробництво, незважаючи на відомі ризики.

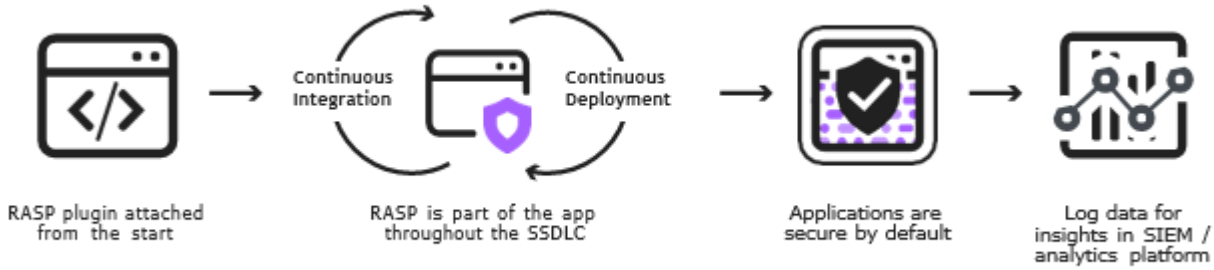

Рис.3.2. Життєвий цикл безпечної розробки програмного забезпечення який узгоджується з автоматизованими процесами DevOps [5]

Таким чином, розробники висувають кілька простих вимог щодо безпеки та контролю:

1. Інструмент безпеки повинен мати незначний вплив на продуктивність програми.

2. Офіс CISO повинен визначити пріоритетність вразливостей за критичністю та надати реальні дані про атаки з виробничого середовища (на відміну від теоретичного аналізу коду).

3. Оскільки офіс CIO впроваджує архітектурні зміни, щоб перейти до програмно-визначеного центру обробки даних, він повинен зробити будь-які зміни інфраструктури прозорими для додатків.

Defenders. Команди зазвичай знаходяться в організаціях CSO/CISO. Їх мета — комплексне вирішення проблем безпеки підприємства та зниження ризиків. Історично до сфери діяльності цих команд входило управління програмами тестування на проникнення та рішеннями для тестування безпеки додатків, а також нагляд за технологіями WAF і RASP. Поки мають доступ до технологій SIEM, більшість аналітичних засобів все ще є надто базовими, щоб надати практичні дані або значно покращити захист та реагування. Керівники інформаційної безпеки, які контролюють роботу захисників, стикаються з чотирма основними перешкодами:

1. Додатки є розподіленими та складними, а підходи до безпеки додатків також фрагментовані. Віртуалізація, мікросервіси та хмара роблять додатки та дані повсюдними, їх неможливо контролювати послідовно й точно. А стратегії безпеки

— це, в основному, лише набір інструментів і послуг ринку, які часто не є гнучкими або портативними.

2. Обсяги Attacfi збільшуються, і складність загальних векторів Attacfi здійснюється шляхом передачі існуючих елементів керування. Потрібні прості, прагматичні засоби керування безпекою додатків, які працюють, щоб запобігти поширеним, сучасним векторам атак, таким як ті, які перераховані в OWASP Top 10, на які припадає більшість атак.

3. Критичні виробничі додатки піддаються ризику через неушкоджені вразливості. Захисники розуміють неминучий ризик, який виникає, коли розробники змушені запускати програми з відомими вразливими місцями. Засоби контролю безпеки та зменшення вразливостей повинні відповідати DevOps, але це не так.

4. Немає видимості подій безпеки під час виконання. Незважаючи на аналіз на етапі попереднього виробництва або розвідки, зібрану на мережевому рівні, Defenders не може отримати доступ до корисних даних про атаки під час виконання.

Розглянемо пропозиції щодо посилення інформаційної системи організації за допомогою технологій виконання для захисту додатків

Gartner вперше визначив самозахист додатків під час виконання (RASP) як технологію безпеки, вбудовану або пов’язану з середовищем виконання програми для контролю виконання та запобігання атак у реальному часі. До того, як RASP вийшла на ринок безпеки, пропозиції галузі забезпечували захист на мережевому рівні та на хості, але не мали активний захист на прикладному рівні. За винятком WAF, не було засобів захисту виробничого середовища для забезпечення контролю під час виконання. Простір безпеки програми не є тривіальним і не чітко визначений.

RASP — це інструмент, який зазвичай підпадає під категорію «Тестування та захист під час виконання» або «Самозахист програми». Національний інститут стандартів і технологій (NIST) навіть визначає самозахист додатків під час виконання (RASP) як засіб контролю для пом'якшення ризику через вразливості

програмного забезпечення. RASP важливий не тільки за своєю власною функцією, але й у тому, як він відрізняється від інших технологій в інформаційній системі та/або взаємодіє з ними. Найчастіше RASP доповнює і навіть покращує ефективність інших засобів. Наступна таблиця має на меті визначити різні технології в інформаційній системі безпеки додатків, які потенційно взаємодіють з RASP, і яким чином.

3.2. Застосування технології RASP в ролі доповнення WAFS

Брандмауер веб-додатків (WAF) зазвичай розташовано перед веб-програмами, перевіряючи вхідний трафік HTTP-запитів на наявність відомих корисних даних атак і ненормальних моделей використання. Коли виявлено підозріле корисне навантаження або використання, WAF може або повідомити про порушення, або повідомити та заблокувати запит. WAF – чудовий варіант для захисту від наступних типів атак: об’ємна (DDoS), автоматизація ботів (збирання, скрейпінгу тощо), поглинання облікового запису (заповнення облікових даних) та безпека API.

RASP пропонує критичний рівень запобігання атак за WAF або WAF наступного покоління. Підхід RASP відрізняється від традиційних WAF, оскільки він тісно пов'язаний з кодом програми. RASP використовує контекстну обізнаність, без чорних або білих списків, для виявлення загроз і забезпечення гарантії того, що конкретне корисне навантаження не зможе використати невідому частину коду програми. Технологія RASP перевіряє повні (і часто перетворені дані) в контексті того, як програма буде їх використовувати, якщо і тільки якщо програма намагатиметься використати дані. Результатом є дуже низький рівень помилкових спрацьовувань і висока видимість вразливостей, включаючи слабкі місця, раніше невідомі організації. Таким чином, RASP доповнює WAF і служить останньою лінією захисту в безпеці програм.

Таблиця 3.1.

RASP як додаткова технологія

Технологія	Взаємодія
Брандмауер (WAF) АБО WAF наступного покоління	WAF контролюють і блокують трафік, застосовуючи правила, і корисні як перший рівень захисту. WAF є хорошим варіантом на периметрі для зупинки DDoS-атак, шкідливих ботів, скриптових дітей тощо. RASP не є проксі-сервером і не блокує трафік; натомість він нейтралізує шкідливі або неправильно сформовані корисні навантаження та конкретні вхідні дані, щоб служити останньою лінією захисту. Оскільки він знаходиться в програмі, він має доступ до деталей атаки, включаючи несанкціоновану активність бази даних. Провідні рішення від виробників WAF розглядають RASP як додаткову технологію..
Динамічне тестування безпеки програми (DAST)	Провідні рішення DAST, як Veracode, забезпечують видимість уразливостей. Взаємодія між DAST і RASP проста. RASP можна використовувати для визначення пріоритетності вразливостей перед запуском будь-якого інструменту тестування, вказуючи розробникам, як найкраще мінімізувати ризик, і забезпечуючи ефективні методи безпечного кодування. RASP також можна встановити на програму, яка працює, і ввімкнути в режимі захисту. Атаки та ненормальні введення очищаються та пом’якшуються в режимі реального часу, а докази захисту від загроз під час виконання повідомляються в журналах та на панелях у провідних SIEM, включаючи Splunk.

Управління інцидентами безпеки та подіями (SIEM)	RASP знаходиться всередині програми та збагачує дані про атаки критичною інформацією про те, де відбулася будь-яка трансформація або спроба ексфільтрації, і значно покращуючи аналітика безпеки SIEM з аналізом часу виконання. Більшість провідних постачальників, таких як Splunk, дозволяють візуально відображати журнали та файли, а також використовувати механізми аналізу даних для визначення шаблонів. Усі рішення RASP повинні мати можливість генерувати моніторинг і журнали захисту у форматах LEEF і JSON. Формат LEEF використовується SIEM, включаючи ArcSight, QRadar і Nitro. Журнали JSON можуть прийматися більш сучасними/гнучкими SIEM, такими як Splunk або навіть Elasticsearch.

3.3. Технологія RASP для пом’якшення вразливості в режимі реального часу від атак на ланцюг поставок

Зусилля по виправленню не можуть перевірити та пом’якшити на 100% вразливості безпеки додатків, виявлені в життєвому циклі розробки безпечного програмного забезпечення. Тим не менш, підприємства часто запускають програми з відомими вразливими місцями, які неможливо усунути через відсутність доступу до бази коду, застарілих фреймворків та інших перешкод. Ці процедури виключення вразливості можуть бути дорогими та надзвичайно ризикованими.

Реалізації RASP мають унікальне положення, щоб допомогти підприємствам захищати програми під час виконання, нейтралізуючи відомі вразливості та захищаючи від раніше невідомих загроз і атак нульового дня. Залежно від

характеру розгортання, RASP може також трансформувати або блокувати запити вмісту та бази даних, щоб усе, що обробляє програма, було безпечним.

Багато організацій використовують RASP для вбудовування останньої лінії захисту, яка переміщується разом із програмою, незалежно від того, чи є вона в хмарі, локально, на стадії підготовки чи у виробництві. Як автоматизований технічний контроль за відповідністю вимогам, RASP знімає тиск розробки, виконуючи пом'якшення вразливості в режимі реального часу.

Технологія RASP масштабована, зручна для розробки безпека додатків.

До RASP безпека додатків і DevOps часто були в суперечності. Все більш поширений, гнучкий характер розробки та розгортання додатків зробив запобігання злому ще складнішим. Віртуалізація, контейнеризація, мікросервіси та хмара роблять програми та дані повсюдними, їх неможливо послідовно й точно відстежувати на різних платформах і середовищах.

Що ще гірше, вимоги до тестування перед виробництвом створюють вузькі місця в процесі розробки програмного забезпечення. Деякі вразливості можуть навіть блокувати випуски, що може бути проблематичним у циклі DevOps швидкого випуску безперервної інтеграції / безперервного розгортання (CI/CD).

RASP може бути впроваджено безпосередньо в процес розробки/розгортання DevOps, щоб програми можна було безпечно розгортати у виробництво без будь-яких затримок. Функції виявлення та запобігання RASP можуть бути вбудовані безпосередньо в кожен випуск програми як частина автоматизованого конвеєра CI/CD, що означає, що програми можуть автоматично захищатися незалежно від того, де вони знаходяться в SSDLC. Тести безпеки більше не прив'язані до розкладів випуску, а виправлення є пріоритетним за допомогою даних про атаки на виробництво.

Технологія RASP надає видимість прихованих атак під час виконання.

Групи реагування не мають уявлення про події безпеки додатків у виробництві, і, таким чином, не можуть точно співвіднести результати вразливості перед виробництвом з даними про атаки під час виконання. Крім того, існує ще більш обмежена видимість спроб доступу або ексфільтрації для програм і баз

даних, які переміщуються в хмару. Існує також значна кількість шуму, створеного результатами тестування інструментів, активністю брандмауера програми та звітами про вразливості. RASP може допомогти операціям з безпеки та командам розробників додатків фільтрувати шум і краще розподіляти ресурси за допомогою інтелекту під час виконання.

RASP надає корельовані журнали безпеки мережі, додатків і баз даних для розумніших, швидших відповідей і потужної видимості фактичного ризику організації під час виконання. Моніторинг безпеки додатків за допомогою RASP – це нова можливість було розроблено, щоб дати підприємствам можливість визначати, які програми насправді піддаються атаці в режимі реального часу (і яким чином), ефективно покращуючи управління ризиками та зусилля з відновлення. Якщо коротко, моніторинг додатків через RASP відповідає на наступні запитання як це показано на рис 3.3.

Рис.3.2. Відповіді на моніторинг через RASP

Зокрема, нова аналітика та цифрова криміналістика на рівні програми також можуть виявити шахрайство з аутентифікацією, авторизацією та транзакційними шахрайствами. Детальна інформація про всі запити до бази даних, що надходять певними програмами, дає змогу детально вести аудит і підтримувати аналіз першопричин для порушень даних.

3.4. Рекомендації використання сценаріїв технології RASP

Для використання сценаріїв технології RASP надамо рекомендації щодо використання RASP щоб точніше продемонструвати, як адміністратори безпеки,

команди DevOps та розробники повинні використовувати RASP для вирішення деяких проблем, з якими сьогодні стикається безпека додатків.

1. Зменшення уразливості відставання (Backlog)

Інструменти тестування безпеки додатків (AST - Application security testing) допомагають виявити вразливості на етапі попереднього виробництва. Але на більшості підприємств відставання з часом збільшується. Програми виконують важливі бізнес-функції та транзакційні функції та часто запускаються у виробництво з відомими вразливими місцями. З RASP більш ніж 95% відставання, можливо, не потрібно буде виправляти або виправляти розробниками, оскільки RASP може нейтралізувати загрозу в разі атаки у виробництві. Це призводить до значного підвищення ефективності, а також до економії ресурсів і витрат.

2. Швидкий випуск додатку який масштабований DevSecOps

Безпека додатків і життєвий цикл розробки програмного забезпечення часто суперечать. Якщо RASP встановлюється безпосередньо в автоматизовану воронку DevOps за допомогою інструментів безперервної інтеграції та безперервної розробки, RASP забезпечує безпеку більш бездоганно, тому елементи керування завжди «вкладаються» у кожен випуск за замовчуванням. Організації можуть швидше запускати програми, не турбуючись про вразливості безпеки. Ця можливість зменшує операційні тертя та сприяє більшій довірі та співпраці між командами, допомагаючи досягти узгодженості та плавності SecDevOps / DevSecOps.

3. Видимість виробничих атак у реальному часі

Впровадження рішення RASP дозволяє отримати повну видимість атак у реальному часі (на відміну від потенційних відомих уразливостей). Ці розширені дані про загрози під час виконання можна надсилати до SIEM та інструментів ведення журналу, щоб інформувати як розробників, так і інші екосистемні продукти, такі як WAF або брандмауери наступного покоління. Це також усуває шум від інших інструментів попереднього виробництва, відкриваючи лише критичні події безпеки під час виконання – відповідаючи на такі запитання: «Які виробничі програми/активи є найбільш атакованими?», «Які події вилучення даних

виникли ззовні брандмауера?» Завдяки покращеній цифровій експертизі та аналізу загроз, операції з безпеки можуть точніше співвідносити вразливості та спрямувати зусилля по виправленню для покращення розвитку та відповідності.

4. Забезпечення інтелект для DevOps

Окрім надання інформації про те, які програми є найбільш безпечними/небезпечними, RASP може надати критично важливі дані для команд DevOps. Подібно до інших інструментів, які розробники використовують на етапі проектування/тестування (наприклад, NewRelic для продуктивності і DAST/SAST для сканування коду), RASP може використовуватися для забезпечення видимості того, що програма буде робити під час виконання (наприклад, виклики бази даних, читання/запис файлів , вхід/вихід, невдалий вхід, бічні виклики з виробничих програм, використовувані версії та фреймворки тощо).

5. Захист від застарілих програм та програм за кодом третіх сторон

На підприємствах, де додатки є бізнесом, захист застарілих програм, які приносять прибуток, є важливою вимогою. Більшість із цих застарілих програм написані старішими мовами, у кількох версіях/екземплярах і не мають активної розробки чи підтримки для виправлення вразливостей. RASP захищає ці програми без потреби розробників або підтримки. Крім того, RASP захищає вразливості програмного забезпечення з відкритим вихідним кодом і стороннього коду без необхідності змінювати код.

6. Остання лінія захисту в моделі безпеки

Якщо WAF або брандмауер наступного покоління є першою лінією захисту, рішення RASP — це рішення остання лінія оборони. Сьогодні програми переважно покладаються на зовнішній захист, як WAF або IPS (системи запобігання вторгненням), і існує потреба вбудовувати функції безпеки в програму, щоб вона могла захистити себе під час виконання. Додаток, оснащений RASP, є потужним, оскільки RASP має бачення логіки, поведінки та виконання програми.

RASP є невід'ємною частиною середовища виконання програми. Вона може виявити спробуви записати великий обсяг даних у пам'ять часу виконання програми або виявити несанкціонований доступ до бази даних. RASP має

можливість в режимі реального часу виконувати такі дії, як припинення сеансів, підвищення попереджень тощо. Ці можливості роблять RASP відмінним доповненням до існуючих WAF.

7. Оптимізація безпеки життєвого циклу розробки програмного забезпечення (SSDLC)

Підприємства повинні продовжувати використовувати технології динамічного та статичного тестування та завершити життєвий цикл безпечної розробки програмного забезпечення (SSDLC), захищаючи свої програми за допомогою RASP під час виробництва. Атаки з підтримкою RASP покращать та оптимізують ефективність SSDLC, щоб забезпечити чітке розподілення ресурсів і чіткі віхи для усунення.

Наприклад, плагіни RASP можуть бути ефективною частиною проактивної програми безпечного кодування. Навчальні програми для розробників можуть позитивно вплинути на кількість уразливостей, виявлених під час кодування, але часто буває складно дати просту відповідь на запитання: «Що я маю виправити? Чи використовується підхід з білим або чорним списком? Які символи слід виключити чи дозволити у вхідних даних? Що з кодуванням? Заплутані дані? Спектакль?" Простою і одноманітною відповіддю було б виправлення критичних вразливостей, які схильні (або були виявлені) експлойту в робочому середовищі, і включення модуля RASP, що підключається, в пакет розгортання програми для зниження ризику вразливостей, що залишилися під низький рівень. Будь-який підхід легко реалізувати в ко без істотних змін у тому, як програма пишеться або тестується.

8. Вдосконалення операції з безпеки і реагування

RASP надає командам центрів безпеки (SOC) інформацію на рівні додатків для прийняття більш швидких, простих та розумних рішень, скорочення життєвого циклу розслідування та покращення контролю за периметром. RASP може об'єднувати інформацію про безпеку мережі, додатків та баз даних у попередньо корельованому звіті, дозволяючи діяти на основі фактичного (а не теоретичного) ризику, наприклад, попереджувальне блокування IP-адрес «зловмисників» без ризику помилкових спрацьовувань.

Традиційно при виникненні події елементи управління периметром надають дані про вихідну IP-адресу, цільову IP-адресу і корисне навантаження, що викликало сигнатуру. Тоді команді безпеки доведеться витратити значну кількість часу на перевірку та тестування, щоб відповісти: Атака реальна? Як ми маємо реагувати?» RASP може створювати звіти та візуалізації для подій у реальному часі з надзвичайно низьким рівнем помилкових спрацьовувань, передаючи дані про атаки в реальному часі у SIEM або інший інструмент ведення журналу, виявляючи, коли база даних повертає аномальні набори даних. Це дозволяє менеджерам з безпеки візуалізувати загрози додаткам у виробничому середовищі та зіставляти їх з іншими джерелами даних, що дозволяє скоротити час, що витрачається на розслідування та аналіз нижчого рівня, натомість дозволяючи відповідним командам швидко реагувати. До RASP менеджери SOC було неможливо зіставляти попереднє виявлення вразливостей із даними про атаки під час виконання. Розуміння способів обходу та переміщення експлойтів між додатками та базами даних може допомогти групам безпеки вживати більш обґрунтованих заходів для більш швидкої цифрової експертизи для виявлення шахрайства з автентифікацією, авторизацією та транзакціями.

9. Захист для програм у будь-якому місці

Архітектура додатків розвивається. Сучасні постачальники послуг на вимогу та інфраструктури як послуги (IaaS) набирають популярність, оскільки вони дають змогу отримати виграш для бізнесу для кращої співпраці та безпеки. Наприклад, віртуалізація та міграція до хмарних сервісів підвищують ефективність безпеки інфраструктури за рахунок створення високопродуктивних кластерів продуктивності та зручного використання. Таким чином, програми та їхні дані зараз повсюдно, їх неможливо відстежувати послідовно та точно. Однак це означає, що безпека має бути гнучкою і портативною. Вона повинна бути сумісною не тільки зі старими та новими мовами програмування, але також з фреймворками веб-додатків і мікросервісами, підтримкою локальних, хмарних і контейнерних розгортань, а також мати пряму інтеграцію з широким набором сканерів коду, інструментів реєстрації даних, і SIEM.

RASP може забезпечити видимість спроб доступу або ексфільтрації для додатків і баз даних, які переміщуються в хмару та між мікросервісами. RASP живе і подорожує в програмі та реєструє всі події безпеки під час виконання. Активність бази даних відстежується з програми, щоб отримати повну інформацію про поведінку на рівні програми. Програми залишаються захищеними незалежно від того, де вони знаходяться, а виробничі дані можна аналізувати в SIEM і використовувати для покращеного моніторингу активності бази даних.

10. Зниження ризиків AppSec і підвищення відповідності дотримання

RASP може служити компенсуючим засобом для неусунених (або непереборних) уразливостей, які в іншому випадку зазнали б дорогого процесу виключення відповідності. Часто це відбувається через відсутність доступу до бази коду, застарілих фреймворків та інших перешкод та нестачі часу. При відповідному захисті під час роботи та залежно від конфігурації, відповідність PCI може бути досягнута швидким, точним і простим в обслуговуванні способом.

3.5. Застосування технології RASP для відповідності стандартам

RASP може слуговати для виконання вимог відповідності. Залежно від реалізації, RASP може забезпечити критичні технічні засоби контролю для організацій, які намагаються виконати суворі нормативні стандарти безпеки та конфіденційності даних, завдяки кільком ключовим функціям:

• RASP забезпечує механізми перевірки даних для запобігання експлуатації потенційно вразливих конструкцій кодування в програмному забезпеченні. Дані, які надходять у програму та через неї, можуть бути перевірені RASP для захисту від відомих, поширених атак рівня прикладних програм і, залежно від методології, загроз нульового дня.

• Включення RASP під час розробки додатків є простим і послідовним способом реалізації можливостей безпеки та видимості, необхідних у життєвому циклі безпечної розробки як для спеціально створених, так і для стандартних програмних програм. Незалежно від того, чи вони розгорнуті під час процесу SDLC

або DevOps, засоби контролю безпеки RASP можуть переміщуватися разом із програмою і завжди залишатися «увімкненими».

• RASP також реєструє інциденти безпеки та дії користувача (ідентифікація користувача, тип події, дата і час, успіх/збій, походження та назва компонента системи, що постраждав) для покращення розслідування та останніх кореляцій.

У випадку стандарту безпеки даних індустрії платіжних карток (PCI DSS), RASP може надати автоматичне керування для виконання ряду вимог (див. таблицю 3.2)

Таблиця 3.2.

Відповідність стандарту безпеки даних індустрії платіжних карток

6.3 Безпечно розробляйте внутрішні та зовнішні програмні додатки
6.5 Усунути поширені вразливості кодування в процесі розробки програмного забезпечення
6.6 Переконайтеся, що програми захищені від відомих атак
10.2 Впровадити автоматизовані аудиторські записи
10.3 Запис записів аудиторського сліду
11.4 Використовуйте методи виявлення/попередження вторгнень

3.6. Методи впровадження технології RASP

РАСП в дії: пасивний і активний

Усі технології RASP повинні мати можливість функціонувати в двох різних, але взаємодоповнюючих режимах: моніторинг і захист.

1. У пасивному режимі моніторингу рішення RASP повинно використовувати дуже обмежені ресурси програми, такі як центральний процесор і пам'ять (RAM). Це також повинно додати мінімальну затримку. У режимі моніторингу RASP повинен мати можливість генерувати подібні події журналу, як якщо б він був у режимі активного захисту. Це дозволяє організаціям створювати

або отримувати доступ до аналітичного звіту безпеки або «теплової карти» про те, де реальні атаки вражають програму.

2. У режимі активного захисту рішення RASP все одно має використовувати обмежені ресурси програми для виявлення загроз, автоматично пом'якшуючи атаки в режимі реального часу та запобігаючи ексфільтрації бази даних. Це не повинно вимагати значних ресурсів для налаштування чи конфігурації, або вимагати громіздких наборів правил чи списків визначень. Це повинно додавати мінімальну затримку додатку. Перебуваючи в режимі активного захисту, RASP має генерувати оперативну інформацію про реальні атаки, а також про те, які дії були виконані для нейтралізації шкідливого або неправильного корисного навантаження.

Технічні складові: аналіз та впровадження

RASP має два унікальних технічних компонента: аналіз безпеки плюс реалізація аналітичного процесора на рівні програми. Технічна оцінка будь-якого RASP повинна обговорювати переваги та слабкі сторони кожного компонента як окремо, так і разом.

1. Аналіз загроз застосування

Спосіб виявлення, обчислення та пом’якшення атак безпеки є одним із найважливіших атрибутів RASP, оскільки він впливає на точність і продуктивність, а також на реалізацію. Наприклад, поширеною проблемою контролю безпеки програм є поширеність помилкових спрацьовувань і помилкових негативів, які виснажують ресурси та створюють громіздку кількість шуму.

Чотири основні методології обчислення атаки — це зіставлення шаблонів, евристика, аналіз потоків даних і теоретико-мовна безпека (LANGSEC). Кожне рішення RASP виконує аналіз загроз, використовуючи інший підхід (а іноді і комбінацію). Короткий огляд кожної з чотирьох методик та їх чутливість до відповідних сповіщень показано далі.

Таблиця 3.3.

Методологія аналізу загроз застосування

Методика	Опис	Хибнопозитивні	Хибнонегативні
Зіставлення шаблонів	У відповідності з шаблоном використовуються рядкові літерали та регулярні вирази, щоб визначити, чи є корисне навантаження безпечним. Приклад: перевірка, чи містить SQL-запит символи коментарів, як-от «.»	Високий Сліпе додавання моделей атак може викликати програми для випадкового зламу та виявлення помилкових спрацьовувань. Шаблони атак повинні оцінюватися розробниками та командами QA вручну та за допомогою автоматизованих тестів, які опрацьовують всю програму.	Високий Якщо моделі атак не існують у корпусі, то результат буде помилково негативним. Важливо ретельно дивитись за шаблоном.
Евристика	Евристика використовує багатокритеріальний аналіз для статистичного	Високий Виявлення аномалій має високу схильність зупиняти відомий	Високий Виявлення аномалій на основі статистичних методів можна порушити,

	визначення проблем без їх визначення. Приклад: виявлення аномалій у життєвих циклах запитів/відповідей HTTP, пошук аномальних подій.	хороший трафік. Наприклад, періодичні заплановані внутрішні завдання (тобто cron), які отримують доступ до веб-сервісів, виходять за межі «звичайних» меж і вимагають спеціального білого списку IP-адрес.	збільшивши вибірки поганої поведінки. Оскільки трафік атаки не позначається як безпечний чи шкідливий через контрольоване машинне навчання, корисні навантаження з високими частотами пройде (нормальна поведінка).
Аналіз потоків даних	Аналіз потоку даних використовує надані мовою інструментальні API для відстеження змінних і дані проходять через програму. Приклад: спостереження за змінними HTTP, які потрапляють у запити SQL.	Високий/Помірний Аналіз потоку має здатність створювати висок кількість помилкових спрацьовувань через різноманітність способів розробки програм. Як приклад, практика	Помірний Аналіз потоку може генерувати сучасні помилкові негативи. Наприклад, якщо програму створено, де інформація протікає по-різному (наприклад, поза смугою черг виробника/споживача), тоді аналіз потоку пропустить атаку.

		внутрішнього розвитку для інтерполяція змінних може не бути справжньою проблемою безпеки.	
Теоретико-мовна безпека (Language security)	LANGSEC – це процес формального розуміння як такі дані, як корисне навантаження вмісту, запити до бази даних, команди операційної системи тощо, будуть виконуватися в середовищі. Приклад: розуміння того, чи містить запит до бази даних тавтологію, суперечність або спробу отримати доступ до недійсного стовпця.	Низька Оскільки LANGSEC спирається на побудову формальних граматик мов, кількість помилкових спрацьовувань значно зменшується.	Низька Оскільки LANGSEC спирається на побудову формальних граматик мов, кількість помилкових негативів значно зменшується.

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи було отримано наступні результати:

Було проведено аналіз щодо ланцюга поставок організацій з використання інформаційної системи. В результаті було отримано основні етапи роботи інформаційної системи SCM, яка дозволяє покращити та пришвидшити роботу бізнес-процесів організації.

Як результат, надалі було отримано основні ланцюжки постачання програмного забезпечення, та визначено проблему щодо впливу атак ланцюга поставок на роботу інформаційної системи в цілому. Тому розробка системи кібербезпеки даних систем повинна покривати атаки OWASP топ-10 і більше.

Для визначеної проблеми ланцюга поставок в інформаційній системі організації було досліджено технологію створення каталогу конкретних шаблонів атак, оцінки загроз та аналізу відновлення щодо виявлення та оцінки кіберзагроз та загроз ланцюга поставок та вибору ефективних контрзаходів в процесі життєвого циклу системи SCRM.

В результаті роботи досліджено технологію Imperva RASP, яка захищає програми, середовище виконання, сервери, залежності з відкритим вихідним кодом та сторонні бібліотеки. Отримано структуру архітектури підприємства, яке використовує веб-додатки для бізнесу, відповідно до якого визначено життєвий цикл безпечної розробки програмного забезпечення, запропоновано застосування технології RASP в ролі доповнення WAFS та створено шляхи подолання атак на ланцюжок поставок.

ПЕРЕЛІК ПОСИЛАНЬ

1 Ланцюг поставок URL: https://www.sap.com/central-asia-caucasus/products/scm/what-is-supply-chain-management.html?url_id=text-central-asia-caucasus-404-reclink (дата звернення: 02.10.2024).

2. SCM системи URL: https://www.it.ua/knowledge-base/technology-innovation/supply-chain-management-scm (дата звернення: 02.10.2024).

3. Supply-Chain-Risk-Management URL: https://csrc.nist.gov/CSRC/media/Projects/Supply-Chain-Risk-Management/documents/ssca/2017-winter/NCSC_Placemat.pdf (дата звернення: 15.10.2024).

3. Supply-chain-attack Imperva URL: https://www.imperva.com/learn/application-security/supply-chain-attack (дата звернення: 02.12.2024).

4. Web-autAppsec URL: https://www.imperva.com/products/runtime-application-self-protection-rasp/ (дата звернення: 27.10.2024).

5. Загальний огляд RASP URL: https://docs.imperva.com/howto/a12e0c22

6. Supply-chain-management URL: https://www.ibm.com/topics/supply-chain-management / (дата звернення: 02.11.2024).

7. Що таке Supply-chain-attack URL: https://ela.kpi.ua/bitstream/123456789/29790/1/170-173.pdf (дата звернення: 12.11.2024).

8. M. Blackmer. Attacking the Weakest Link in the Supply Chain — Cisco Blog изд. — 2023 URL: https://blogs.cisco.com/security/ attacking-the-weakest-link-in-the-supply-chain? dtid=osscdc000283 (дата звернення: 22.11.2024).

9. P. Moorhead. That Time Of Year Again: Cisco Systems Releases Its Annual Cybersecurity Report — Forbes. — 2018. — URL: https:// www.forbes.com/sites/patrickmoorhead/2018/ (дата звернення: 02.12.2024).

10. Cyber-security risks in the supply chain — Cert-uk вид. — 2021. URL: https://www. cert.gov.uk/wp-content/uploads/2015/02/ Cyber-security-risks-in-the-supply-chain. Pdf (дата звернення: 08.12.2024).

11. R. WAUGH. The terrifying rise of cyber crime: Your computer is currently being targeted by criminal gangs looking to harvest your personal details and steal your money. — Mail Online изд. — 2013. URL: https://www.dailymail. co.uk/home/moslive/article-2260221/ Cyber-crime-Your-currently-targeted-criminal-gangs-lohtml (дата звернення: 10.12.2024).

12. ISO/IEC 27001:2013, Information technology – Security Techniques – Information security management systems – Requirements — ISO вид. — 2013. URL: https://www.iso.org/ isoiec-27001-information-security.html (дата звернення: 10.12.2024).

13. ISO/IEC 28001:2007, Security management systems for the supply chain – Best practices for implementing supply chain security, assessments and plans – Requirements and guidance. — ISO вид. — 2007. URL: https://www.iso. org/standard/45654.html (дата звернення: 10.12.2024).

14. Щибун Є.Ю. Атаки на ланцюг поставок supply chain. *Актуальні проблеми кібербезпеки:* Матеріали Всеукраїнської науково-практичної конференції (м. Київ, 25 жовтня 2024 року). Київ, 2024. С. 208-210. (дата звернення: 05.11.2024).