УДК 621.391

МОДЕЛЬ ЗАГРОЗ У ГЕТЕРОГЕННОМУ СЕРЕДОВИЩІ З ДИНАМІЧНОЮ АРХІТЕКТУРОЮ З НЕВІДЧУЖУВАНИМИ ОБЧИСЛЮВАЛЬНИМИ ВУЗЛАМИ ПІД КЕРУВАННЯМ KUBERNETES

Шенкаренко Т.О., Алєксєєв М.О
Навчально-науковий Інститут телекомунікаційних систем КПІ ім. Ігоря Сікорського, Україна
E-mail: shenkarnekotaras@gmail.com, alexeyev@its.kpi.ua

THREAT MODEL IN A HETEROGENEOUS ENVIRONMENT WITH A DYNAMIC ARCHITECTURE AND INALIENABLE COMPUTING NODES MANAGED BY KUBERNETES

The work is dedicated to the pressing issue of threat modeling in a heterogeneous environment with a dynamic architecture and non-alienable computing nodes managed by Kubernetes. The proposed approach is based on the implementation of improved approaches applied during threat modeling, focusing on process optimization, data collection and analysis, threat contextualization, updating vulnerability information based on public databases, and considering the business context.

Unlike existing known solutions, the proposed enhanced threat modeling approaches include key aspects such as detailed information gathering, supply chain analysis, system compliance verification with data protection regulations (such as the General Data Protection Regulation), and the implementation of best practices from the Open Web Application Security Project (OWASP) [1] at early stages of software development.

The advantage of the proposed approach is the early detection of vulnerabilities, as the enhanced approaches focus on a detailed analysis of the future product's documentation and architecture. This ultimately leads to resource and cost savings in product development since fixing vulnerabilities at the design stage is significantly cheaper than after the product has entered the market. Additionally, it reduces the risk of extra expenses on refactoring, testing, and releasing updates.

The study examines the experimental web environment OWASP Juicy Shop running under Kubernetes, builds a threat model using the graphical tool Threat Dragon, investigates the proposed enhanced methods, provides a statistical comparison with traditional threat modeling methods, and justifies the effectiveness of the proposed approach.

У наш час моделювання загроз стикається з низкою складнощів, пов'язаних із динамічністю IT-інфраструктури, зростанням атак на ланцюги постачання, та посиленням регуляторних вимог. Розвиток хмарних технологій, використання веб-додатків під управлінням Kubernetes та ускладнення мікросервісної архітектури призводять до розмивання традиційних меж довіри, що ускладнює аналіз потенційних загроз. У цьому контексті традиційні підходи до моделювання загроз, такі як STRIDE, P.A.S.T.A. та аналіз вразливостей через тестування на проникнення, все ще можуть бути корисними, де інфраструктура є менш динамічною і не включає складних інтеграцій із зовнішніми компонентами чи мікро сервісами, які зараз характерні для Kubernetes, однак вони вимагають адаптації до нових умов.

Для забезпечення безпеки програмного продукту й даних користувачів, у роботі розглядаються інструменти та пропонується покращена стратегія для моделювання загроз, яка повинна використовуватися в ранніх етапах SSDLC (Secure Software Development Lifecycle). Порівняно з традиційними, покращена стратегія включає у себе:

1. Аналіз потенційних вразливостей ланцюгів постачання. Згідно з запропонованою покращеною стратегію, необхідно визначити всі компанії та постачальників, які є частиною ланцюга постачання, включаючи не тільки основних постачальників, а й третіх осіб, таких як підрядники, субпідрядники та інші партнери. Важливо мати повний список усіх потенційних точок доступу, через які можуть бути здійснені атаки. Для аналізу відомих вразливостей в ланцюгах постачання, пропонується використовувати наступні сервіси: CVE (Common Vulnerabilities and Exposures) – це база даних, яка містить інформацію про публічно відомі уразливості в програмному забезпеченні та апаратному забезпеченні; NVD (National Vulnerability Database) – національна база даних вразливостей, яка є частиною федерального уряду США. Вона надає розширену інформацію про вразливості, їх вплив на безпеку та технічні деталі, що корисно для аналізу систем та програм, що використовуються постачальниками; Snyk – інструмент для автоматизованого сканування відкритих кодів на наявність вразливостей. Особливо корисний для оцінки безпеки у програмному забезпеченні та бібліотеках, які використовуються сторонніми постачальниками.

2. Більш глибокий збір інформації для оцінки потенційних векторів атак. Комплексний збір даних включає в себе інформацію про потоки даних, конфігурації системи, аналіз YAML-файлів маніфестів, журналів Kubernetes, мережевої активності та архітектури сервісів, тощо.

3. Інтеграція інструментів OWASP у процес моделювання. Запропонована стратегія використовує інтеграцію методології й інструментів OWASP у процес моделювання, таких як:

- OWASP Dependency-Check. Цей інструмент дозволяє здійснювати аналіз залежності й виявляти відомі уразливості в сторонніх бібліотеках та компонентах, які використовуються в додатку. В контексті Kubernetes, де додатки часто використовують контейнеризацію та інтеграцію з зовнішніми сервісами, аналіз залежності допомагає запобігти атакам через уразливості в цих компонентах.

- OWASP ASVS (Application Security Verification Standard) – це набір стандартів і рекомендацій, які визначають вимоги до безпеки веб-додатків та API

- OWASP Threat Dragon — сучасний інструмент для моделювання загроз, який забезпечує інтерактивне створення діаграм потоків даних та аналіз потенційних вразливостей у системах.

4. Урахування бізнес-контексту під час моделювання. Безпека системи повинна враховувати не лише технічні аспекти, але й особливості її бізнес-середовища. Це включає розуміння ключових вимог бізнесу, які впливають на проектування, функціональність, підхід захисту системи та відповідність щодо

вимог регуляторів [3], таких як: GDPR (General Data Protection Regulation) — захист персональних даних громадян ЄС, включаючи вимоги до отримання згоди та права на видалення даних. PCI DSS (Payment Card Industry Data Security Standard) — захист даних платіжних карток. ISO/IEC 27001 — міжнародний стандарт управління інформаційною безпекою. HIPAA (Health Insurance Portability and Accountability Act) — це федеральний закон США, ухвалений у 1996 році для регулювання захисту конфіденційності та безпеки медичної інформації.

Таким чином, процес моделювання набуває більш актуального сенсу у контексті раннього виявлення загроз й слабких місць архітектури продукту. В таблиці 1 представлено порівняння традиційних стратегій і запропонованої покращеної стратегії такого моделювання.

Таблиця 1. Порівняння традиційних і запропонованої покращеної стратегії моделювання загроз й слабких місць архітектури продукту.

Аспект	**Традиційні стратегії**	**Покращена стратегія**
Аналіз вразливостей ланцюгів постачання	Зазвичай обмежується лише основними постачальниками	Визначення всіх учасників ланцюга постачання, включаючи третіх осіб (підрядники, субпідрядники). Використання баз даних CVE, NVD, Snyk для виявлення відомих вразливостей.
Збір інформації для оцінки векторів атак	Обмежений збір інформації за допомогою стандартних журналів безпеки.	Глибокий збір інформації про потоки даних, конфігурації системи, YAML-файли маніфестів Kubernetes, мережеву активність та архітектуру сервісів.
Інтеграція інструментів OWASP	Може бути обмеженою або взагалі не використовуватися	Інтеграція таких інструментів, як OWASP Dependency-Check для аналізу залежності, OWASP ASVS для стандартів безпеки, OWASP Threat Dragon для інтерактивного моделювання загроз.
Аналіз безпеки на етапі проектування	Зазвичай обмежений лише технічними аспектами без урахування бізнесу.	Урахування не лише технічних аспектів безпеки, але й бізнес-контексту, включаючи вимоги регуляторів, таких як GDPR, PCI DSS, ISO/IEC 27001, HIPAA.
Аналіз потенційних атак на контейнеризацію	Часто не розглядається, якщо не є специфічним завданням.	Оцінка загроз для контейнеризованих додатків, зокрема Kubernetes, включаючи моніторинг контейнерів, мережевих налаштувань і доступу до внутрішніх сервісів
Регулярні перевірки та оновлення	Перевірки здійснюються після впровадження рішення, рідше в процесі розробки.	Постійний процес перевірок і оновлень, зокрема через публічні бази даних вразливостей, для виявлення нових вразливостей та актуалізації моделей загроз.

Для перевірки запропонованої покращеної стратегії моделювання загроз для Web середовища під керуванням Kubernetes було обрано проект OWASP Juice Shop, що прибирає ризики нанесення шкоди реальному середовищу й дає доступ до усієї технічної документації проекту. Це навчальний веб-додаток з відкритим кодом розроблений проектом й його основна мета — навчання розробників, фахівців інформаційної безпеки та інших спеціалістів, знаходити й усувати вразливості у веб-додатках. Цей сервіс спеціально створений для проведення практичних занять і хакатонів з кібербезпеки. Особливістю цього продукту є реалістичність, тому що він містить широкий спектр вразливостей, які зустрічаються в реальних веб-додатках, включаючи вразливості з OWASP Top-10.

Для заповнення даних моделі було зауважено наступні пункти.

1. *Причина*: визначити охоплені та не охоплені вектори атак і потенційні слабкі місця.

2. *Об'єкт оцінювання*: веб-додаток під керуванням Kubernetes, включаючи стандартний функціонал користувача й панелі адміністратора.

3. *Обсяг*: функціональність веб-додатка, зокрема механізми аутентифікації, авторизації, обробки даних користувачів та платежів.

Взаємодія з клієнтськими пристроями та зовнішніми сервісами (наприклад, платіжними шлюзами) не входить в обсяг оцінювання й моделювання.

Експериментальне моделювання загроз мало наступні фази:

1. Аналіз точок входу для ймовірної атаки на вузли системи.
2. Аналіз точок виходу.
3. Перерахунок активів системи.
4. Вектори атак для ідентифікованих активів.
5. Механізми пом'якшення ризиків.

Після заповнення всіх пунктів, було проведене графічного моделювання у програмі OWASP Threat Dragon, с заповненням вузлів архітектури й потенційні векторів атак.

У таблиці 2 приведені результати моделювання загроз для Web середовища. Дані для традиційних стратегій взяті з практичного досвіду та аналізу відкритих джерел [4].

Таблиця 2. Результати моделювання загроз.

Параметр	**Традиційні стратегії**	**Запропонована стратегія**	**Різниця**
Кількість вірно ідентифікованих загроз	70	85	21%
Точність моделювання	82%	100%	18%
Час витрачений на моделювання	Менший (31 години)	Більше (49 годин)	16%
Відповідність регулятивним вимогам згідно GDPR	Середній	Висока	

Нижче приведені розрахунки з таблиці 2, які включають точність моделювання за традиційними стратегіями для веб-сервісу OWASP Juice Shop під керування Kubernetes. Для традиційних стратегій моделювання загроз для платформи OWASP Juice Shop можна прийняти наступні дані, виходячи з типових характеристик процесу моделювання загроз без використання надмірно складних або автоматизованих методів, та загально доступних звітів різних компаній, котрі займаються кібербезпекою:

- кількість вірно ідентифікованих загроз: 70 загроз;
- кількість пропущених загроз: 10 загроз;
- кількість неправдивих позитивних результатів: 5 загроз;

$$Точність\ моделювання = \frac{70}{70 + 10 + 5} \times 100\% \approx 82{,}35\%$$

Оскільки за запропонованою стратегією немає пропущених загроз й неправдивих позитивних результатів, точність моделювання для веб-сервісу OWASP Juice Shop під керуванням Kubernetes буде дорівнювати 100%. Кількість вірно ідентифікованих загроз: 85 загроз.

Таким чином, різниця у точності моделювання за традиційними й запропонованою стратегією 100 - 82,35 = 17,65 %

Висновки. Завдяки покращеній стратегії моделювання, кількість вірно ідентифікованих загроз зросла на 21%, при різниці додатково витраченого часу у 16%, але моделювання відбувалось хоча і на реалістичних, але все ж таки синтетичних даних, у реальному житті цей показник може дещо відрізнятись.

Модель включає у себе вимоги регулятора GDPR, що знижує ризики на додаткове навантаження бюджету при розробці програмного продукту у майбутньому.

На відміну від приведених традиційних стратегій, покращена стратегія враховує оцінку загроз для контейнеризованих додатків, зокрема Kubernetes.

Література

1. OWASP Application Security Verification Standard. Режим доступу: https://owasp.org/www-project-application-security-verification-standard
2. AWS. "Security in Containers and Kubernetes on AWS." Режим доступу: https://aws.amazon.com
3. IT Governance Publishing, “EU General Data Protection Regulation (GDPR): An Implementation and Compliance Guide" (4th Edition)” October 15, 2020.
4. Threatmodeler. Continuous cybersecurity in 2024. Режим доступу: https://www.threatmodeler.com/continuous-cybersecurity-in-2024-what-works-and-what-doesnt/