В.В. Семерков,
студент,
Харківський національний університет радіоелектроніки,
м. Харків, Україна
Керівник: кандидат технічних наук,
професор ***О.Г. Качко***

ДОСЛІДЖЕННЯ МЕТОДІВ ТА ПЛАТФОРМ РОЗРОБКИ ЗАХИЩЕНИХ INTERNET ДОДАТКІВ

З розвитком мережі Internet все більше користувачів мають можливість оцінити її переваги. Завдяки їй ми отримали нові можливості у спілкуванні, обміні даними, їх зберіганні та використанні Internet сервісів. В наш час Internet відіграє важливе значення у створенні інформаційного простору глобального суспільства, слугує фізичною основою доступу до веб-сайтів і багатьох систем (протоколів) передачі даних.

Веб-додатки – найбільш поширені програмні сервіси, доступні через Internet, тому вони можуть зазнати небезпеки з боку зловмисників, які можуть викрасти цінну інформацію, зіпсувати дані або якось інакше скомпрометувати систему. Наслідки несанкціонованого доступу та атак стають все більш поширеними і суттєвими.

Уразливості стали цілком звичайним явищем в сучасних веб-додатках. Як приклад можна назвати дефекти безпеки в коді веб-сторінки, які можуть дозволити одному клієнту системи переглядати інформацію іншого. Такі уразливості можуть дозволити виконати запити до серверної бази даних програми і, можливо, отримати контроль над веб-сервером клієнта [1, с. 92]. Уразливості безпеки являють собою неминучу і зростаючу загрозу.

Забезпечення безпеки веб-додатків – серйозне завдання, якому потрібно приділяти належну увагу на всіх етапах – при проектуванні, розробці, розгортанні і експлуатації. Також треба застосувати необхідні засоби захисту, підтримувані платформою. Навіть коли розроблений додаток буде працювати на відносно безпечній платформі, на етапах проектування, розробки і розгортання потрібно використовувати передовий досвід в галузі безпеки, щоб забезпечити максимальний рівень захисту від атак.

На даний час найбільш розповсюдженими та популярними є такі платформи розробки Internet додатків:

– Java EE – найбільш популярна технологія розробки кросплатформових веб-додатків, використовується у високопродуктивних проектах, в яких необхідна надійність, масштабованість, гнучкість;

– .Net – програмна технологія, запропонована компанією Microsoft як платформа для створення як звичайних програм, так і веб-застосунків. Багато в чому є продовженням ідей та принципів, покладених в технологію Java;

– Ruby on Rails – об'єктно-орієнтований фреймворк для створення веб-застосунків, написаний на мові програмування Ruby. Ruby on Rails надає каркас модель-вид-контролер (Model-View-Controller) для веб-застосунків, а також забезпечує їхню інтеграцію з веб-сервером і сервером бази даних;

– Django – високорівневий відкритий Python-фреймворк для розробки веб-систем.

Для цих платформ написана більшість сучасних застосувань, які розгорнуті у мережі Internet та повинні бути захищеними.

Захист відіграє у інформаційних системах надзвичайно важливу роль. Такі системи повинні надавати користувачам і розробникам механізми, що забезпечують реалізацію різноманітних правил захисту. Розробка і правильне застосування подібних механізмів зазвичай роблять забезпечення захисту в інформаційних системах складним інженерним завданням.

Слід виділити три важливих моменти. Перший з них полягає в тому, що інформаційні системи повинні мати засоби для організації захищених каналів зв'язку між процесами. Захищений канал в принципі виконує взаємну аутентифікацію сторін і захищає повідомлення під час пересилання від фальсифікації. Під час аутентифікації встановлюється достовірність твердження, що об’єкт (суб’єкт) має очікувані властивості. Як правило, достовірність твердження встановлюється

з деякою імовірністю. Аутентифікація забезпечує гарантії заявлених ідентифікаційних даних об'єкта та може розглядатися тільки в контексті взаємовідносин між комітентом і об'єктом, який перевіряє [2, с. 206]. Захищений канал зазвичай надає також і засоби підтримання конфіденційності. Це означає, що ніхто, крім зв'язаних один з одним сторін, не в змозі читати повідомлення, що передаються по каналу. Одним із серйозних питань, яке слід вирішити при проектуванні, є вибір між виключно симетричною криптосистемою (заснованою на спільному використанні секретних ключів) або поєднанням її з системою з відкритим ключем. На практиці криптографія з відкритим ключем використовується для розсилки загальних секретних ключів, відомих також як сеансові ключі.

Друге питання захисту інформаційних систем – це контроль доступу, або авторизація. Авторизація стосується захисту ресурсів, щоб тільки процеси, які мають відповідні права доступу, могли отримувати доступ до відповідних ресурсів і використовувати їх. Після аутентифікації процесу завжди проводиться контроль доступу. Кожен з ресурсів може підтримувати власний список доступу, в якому перераховуються права доступу всіх користувачів або процесів [3, с. 536].

Особливу увагу слід приділити питанням управління доступом в разі мобільного коду (також відомого як аплети або завантажувані коди). Двома найбільш розповсюдженими типами мобільних кодів є Java та ActiveX. Крім необхідності захисту мобільного коду від шкідливих хостів, зазвичай важливіше захистити хости від шкідливого мобільного коду. Для цього існують різні способи, з яких найбільш часто застосовується так зване сито. Однак сито надмірно обмежує можливості програм, тому для вирішення проблеми були розроблені більш гнучкі методи на основі реально захищених доменів.

Третє ключове питання захисту інформаційних систем – це управління захистом. Тут є два важливих аспекти – управління ключами і управління авторизацією. Управління ключами включає поширення криптографічних ключів, в якому значну роль відіграють сертифікати, що видаються довіреною третьою особою. При управлінні авторизацією важливі сертифікати атрибутів і делегування.

Крім перерахування векторів атак, ми повинні оцінити ймовірність реалізації тих чи інших загроз і збиток від них, а також виявити фактори, які призводять до виникнення цих загроз. Останній момент дуже важливий: іноді усувати причини виникнення загроз набагато ефективніше, ніж боротися з ними та їх наслідками.

Спочатку потрібно описати систему, яка аналізується, та окреслити її межі. Далі треба займатися ідентифікацією небезпек і попередньою оцінкою наслідків від їх реалізації. Іншими словами, ми повинні оцінити, які загрози можуть бути для системи взагалі, і оцінити наслідки настання кожної із загроз для нашої системи. Далі, якщо наслідки можуть бути серйозними, ми з цією загрозою починаємо боротися в першу чергу, нейтралізовувати її і нівелювати її наслідки. Після цього ми займаємось вже оцінкою величини конкретної загрози – це оцінка ймовірності, і розрахунок підсумкового значення ризику. Потім слід перевірити результати аналізу [4, с. 67]. Процес побудови моделі загроз зображено на рисунку 1.

Далі складається перелік загроз, щоб надалі проаналізувати список і визначити, які з них актуальні. Методів такого аналізу існує досить багато, але універсального серед них немає. При виборі необхідно враховувати велику кількість аспектів. Серед усього різноманіття сценаріїв треба вибрати той, який підходить саме для поставленого при моделюванні загроз завдання.

Починається даний алгоритм з відповіді на питання, на якій стадії знаходиться система, що аналізується: це тільки концепція, робочий проект, система вже експлуатується або мова йде про модернізацію. Треба порівняти альтернативні системи з точки зору числа їх загроз або вибрати заходи щодо зниження цих ризиків? Наскільки складна система – вона проста або багатокомпонентна, складається з багатьох підсистем, або аналізуються загрози тільки для однієї з підсистем? Якого збитку може завдати дана загроза?

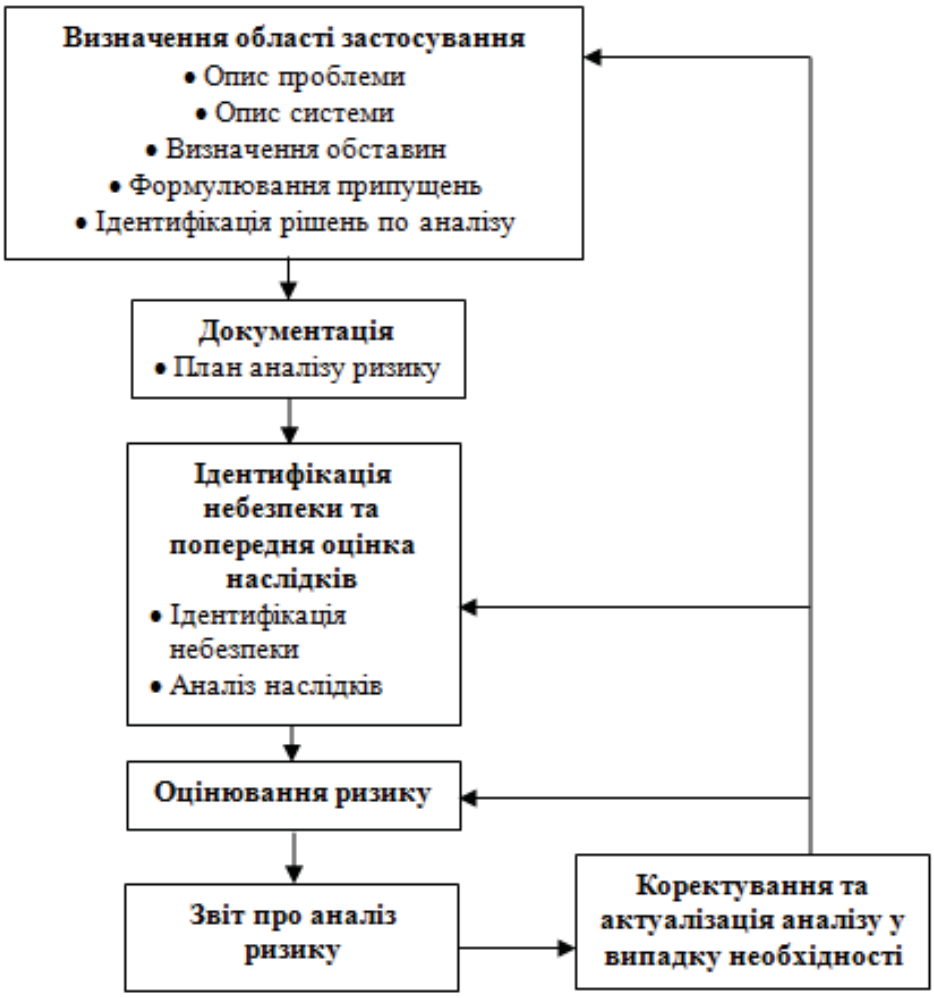

Рис. 1. Процес побудови моделі загроз

Для створення захищених веб-додатків потрібно використовувати специфіки платформи в поєднанні з методиками проектування безпечних систем, моделюванням загроз і тестуванням системи захисту на проникнення. Виконуючи ці базові вимоги можна розробити проект, який буде мати успіх та подальший розвиток.

ЛІТЕРАТУРА

1. Bhargan A. Secure Java / A. Bhargan, B. Kumar. – CRC Press, 2010. – 276 с.
2. Горбенко Ю. І., Горбенко І. Д. Інфраструктури відкритих ключів. Системи ЕЦП. Теорія та практика / Ю. Г. Горбенко, І. Л. Горбенко. – Харків : Форт, 2010. – 593 с.
3. Таненбаум, Э., Ван Стеен, М. Распределенные системы. Принципы и парадигмы / Э. Таненбаум, М. Ван Стен. – СПб. : Питер, 2003. – 877 с.
4. Лукацкий А. Какой должна быть модель угроз // IT-Manager. – 2011. – № 10 – С. 66–69.