УДК 004.492.3 Сучасні інформаційні технології/4. Інформаційна безпека

Антипенко М. С., Ільєнко А. В.

Національний авіаційний університет, Київ, Україна

Методи захисту вебдодатків від вразливостей

Вступ. Вебдодатки мають велике значення у сучасному суспільстві та бізнесі. При розробці вебдодатків використовуються різноманітні технології, які швидко розвиваються та вдосконалюються. Проте використання вебдодатків завжди супроводжується ризиками, які спричиняють зменшення захищеності інформації, що обробляється у вебдодатку. Це обумовлено тим, що зловмисники постійно вдосконалюють засоби і методи, за допомогою яких здійснюються атаки. Найчастіше основною передумовою виникнення вразливостей є недоліки в компонентах вебдодатків, які можуть існувати в них з самого початку їх проектування або з'явитися на подальших етапах розробки та експлуатації.

Актуальність дослідження полягає в тому, що безпека вебдодатків є однією із основних проблем інформаційної безпеки вже понад 10 років. Дійсно, сучасні бізнес-процеси і повсякденне життя - все більше і більше залежать від використання вебдодатків в найрізноманітніших аспектах: від складних інфраструктурних систем до IoT пристроїв. Проте спеціалізованих методів та засобів захисту вебдодатків досить мало, здебільшого це завдання покладають на розробників. Це і використання різних фреймворків, засобів санації, очищення даних, нормалізації і багато чого іншого.

Постановка завдання та мета роботи. Дослідження методів захисту вебдодатків від поширених вразливостей.

Викладення матеріалу. На сьогоднішній день з досліджень OWASP [1] найбільш ефективним способом виявлення вразливостей є експертний аналіз вихідного коду (code review). Огляд коду (Code Review) - один з методів аналізу коду. Суть огляду коду полягає в спільному уважному читанні всього коду програми. В процесі огляду коду тестувальники дають поради в поліпшенні коду. Автор коду при цьому не повинен давати пояснень, як працює та чи інша частина коду, так як вважається що алгоритм повинен бути зрозумілий з коду

або коментарів. При відсутності цих правил код повинен бути допрацьований. Цей спосіб трудомісткий і вимагає високої кваліфікації експерта і не захищений від помилок спеціаліста. Тому активно розвиваються методи автоматичного виявлення вразливостей веб-додатків [2].

Методи автоматичного виявлення вразливостей додатків діляться на дві основні групи:

1. методи, що аналізують роботу вебдодатку без звернення до вихідного коду:

- метод отримання ідентифікуючої інформації про вебдодаток і виявлення його вразливостей за допомогою бюлетенів безпеки (security advisory);
- метод тестування на проникнення – полягає в використанні вебдодатку з точки зору потенційного зловмисника.

2. методи, що аналізують вихідні коди вебдодатків та конфігураційні налаштування:

- метод статичного аналізу коду вебдодатку - аналіз програмного забезпечення, що здійснюється без реального виконання досліджуваних програм. В більшості випадків аналіз проводиться над будь-якою версією вихідного коду, хоча іноді аналізу піддається який-небудь вид об’єктного коду, наприклад Р-код або код на MSIL. Статичний аналіз можна розглядати як автоматизований процес огляду коду.
- метод динамічного аналізу вихідного коду веб-додатку - це тестування і оцінка програми шляхом виконання даних в режимі реального часу. Динамічний аналіз можна поділити на такі пункти:
  - підготовка вихідних даних
  - проведення тестового запуску програми і збір необхідних параметрів
  - аналіз отриманих даних

Тестовий запуск виконання програми можливо відтворювати як на реальному, так і на віртуальному процесорі. Тому для аналізу потрібно з вихідного коду отримати виконуваний файл, але при цьому неможливо проаналізувати код, що містить помилки компіляції або збірки.

Метою динамічного аналізу є знаходження вразливостей безпеки в програмі під час її роботи. На відміну від статичного аналізу, програма динамічного аналізу не має доступу до вихідного коду і тому виявляє уразливості, фактично виконуючи атаки.

Також основними засобами захисту веб-додатків є так звані брандмауери вебдодатків - це особливий тип систем виявлення та запобігання вторгнень, які призначені для захисту веб-додатків [3]. Вони здатні фільтрувати і відстежувати трафік, виявляти або блокувати атаки такі як SQL-ін'єкції або міжсайтовий скриптинг. Вони також мають справу з поширеними неправильними налаштуваннями безпеки веб-сервера і вразливостями, які можуть бути використані зловмисниками. Прикладом популярного брандмауеру з відкритим вихідним кодом є ModSecurity.

Розглянемо детальніше метод тестування вебдодатків на проникнення.

Метод тестування на проникнення розглядає веб-додаток з точки зору зовнішнього користувача, тобто потенційного зловмисника. При цьому вважається, що зловмисник володіє такими ж можливостями, як і звичайний користувач, тобто не має доступу до вихідного коду і доступу до сервера, де знаходиться вебдодаток. Цей метод тестує додаток посилаючи запити, які імітують призначену для користувача активність, що включає «коректні» запити, відповідні нормальним діям користувача і некоректні запити, відповідні діям зловмисника. При пошуку вразливостей в вебдодатку методом тестування на проникнення виникають три основних завдання [4]:

1. Завдання отримання і аналізу структури веб-додатку - суть цього завдання полягає в тому, щоб побудувати повний список URL вебдодатку, методів доступу до нього і списків параметрів, виділити URL, захищені аутентифікацією. Дана інформація необхідна для побудови набору тестових запитів до веб-додатку.
2. Завдання побудови набору тестових HTTP-запитів на основі побудованої структури веб-додатку - суть цього завдання полягає в тому, щоб за вихідними даними (списку URI додатку, методів доступу до кожного URI

і набору параметрів до даного URI) підібрати запити так, щоб було виявлено якомога більше вразливостей.

3. Завдання прогону тестового набору з аналізом відповідей вебдодатку для виявлення вразливостей - це завдання ставить перед собою мету зробити правильний висновок про те, чи демонструє даний HTTP-запит наявність уразливості в веб-додатку чи ні.

Висновок. В ході дослідження було описано необхідність розвитку методів знаходження вразливостей вебдодатків для мінімізації ризиків. На підставі проведеного аналізу можна зробити висновок, що подальший розвиток методів виявлення вразливостей вебдодатків піде по шляху інтеграції можливостей різних методів з тим, щоб було можливо контролювати комплексність виявлення вразливостей різних класів. Це у свою чергу спричинить розвиток автоматичного аналізу, завдяки якому можливо дати гарантію відсутності вразливостей заданих класів.

Література:

1. Curphey M., Wiesman A., Van der Stock A., Stirbei R. A Guide to Building Secure Web Applications and Web Services. OWASP, 2005.
2. Козлов Д. Д., Петухов А. А. «Методы обнаружения уязвимостей в web – приложениях» / Программные системы и инструменты: тематический сборник ф-та ВМиК МГУ им. Ломоносова N 7. П/р Л.Н. Королева. М: Издательский отдел ВМиК МГУ. Изд-во МАКС Пресс, 2006 г.
3. Баранов А. П. «Актуальные проблемы в сфере обеспечения информационной безопасности программного обеспечения, Вопрос кибербезопасности» [Cybersecurity issues], 2015, No 1 (9), pp.2-5.
4. Калашников А.О., Ермилов Е.В., Чопоров О.Н., Разинкин К.А. Баранников Н.И. «Атаки на информационно-технологическую инфраструктуру критически важных объектов: оценка и регулирование рисков». Воронеж: Издательство «Научная книга», 2013. – 160 с.