РОЗРОБКА СИСТЕМИ АНАЛІЗУ БЕЗПЕКИ ВЕБ-ДОДАТКІВ

Зоркіна А.В., Савочкіна А.Ю.
Науковий керівник – ст. викладач., к.т.н. Сінельнікова О.І.
Харківський національний університет радіоелектроніки
(61166, Харків, пр. Леніна, 14, каф. Інформатики,
тел. +38(057)-702-14-19)
E-mail: alionka.zorkina@mail.ru, savockinaa@gmail.com; тел
+380660477086, +380661762657

Кількість програмних продуктів, заснованих на веб-технологіях, з кожним днем збільшується і в даний час стоїть проблема безпеки веб-додатків. Забезпечення безпеки програм є складним процесом в силу великої кількості можливих векторів атак і різноманітності потенційно уразливих компонентів системи. Для вирішення цього завдання потрібно широке тестове покриття, використання спеціальних знань і навичок, а також наявність необхідного інструментарію. У більшості компаній тестування безпеки веб-додатків перевіряється вручну, незважаючи на те, що це дуже трудомісткий процес, який забирає значну частину часу при тестуванні функціональності.

Автоматизація процесу тестування безпеки веб-додатків є одним з пріоритетних напрямків, оскільки дозволяє в короткі терміни зменшити витрати компанії на тестування безпеки, зменшити тривалість процесу тестування і збільшити кількість перевірених сценаріїв.

Згідно з OWASP (Open Web Application Security Project), міжнародної некомерційної організації, зосередженої на аналізі та поліпшенні безпеки програмного забезпечення, найбільш поширеною є проблема недостатньої перевірки вхідних даних. Внаслідок чого можуть бути реалізовані різні атаки. Розглянемо деякі з них.

Ін’єкції (Injections) – всі дані, як правило, зберігаються у спеціальних базах, звернення до яких будується у вигляді запитів, частіше усього написаних на спеціальній мові запитів SQL (Structured Query Language - мова структурованих запитів).

При недостатній перевірці даних від користувача зловмисник може впровадити у форму веб-інтерфейсу програми спеціальний код, який містить частину SQL-запиту. Такий вид атаки називається ін’єкція, в даному випадку самий поширений – SQL- ін’єкція. Це найнебезпечніша вразливість, яка дозволяє зловмиснику отримувати доступ до бази даних і можливість читати/змінювати/видаляти інформацію, яка для нього не призначена. Наприклад, змінити разом з ім’ям і прізвищем баланс свого

рахунку, подивитись баланс чужого рахунку, або викрасти конфіденціальні особисті дані.

Ця вразливість є наслідком недостатньої перевірки даних, які надходять від користувача. Це дозволяє зловмиснику «підсунути», наприклад, в веб-формі, спеціально підготовлені запити, які «обдурять» програму і дозволять прочитати або записати нелегітимні дані. В цілому ця різновидність атак має загальну назву «Помилки валідації», до неї відносять далеко не тільки SQL-ін'єкції.

В даний час існує три підходи виявлення вразливостей веб-додатків, в тому числі ін'єкції: ручний пошук, пошук по шаблонах, fuzzing.

Пошук вразливостей по шаблону – автоматизований метод, заснований на порівнянні деяких характеристик досліджуваних ПЗ з заздалегідь підготовленим описом (сигнатурами) вразливих місць. Даний метод є ефективним для пошуку вразливостей і немаскованих закладок, таких як переповнення буферу, парольні константи, тощо.

Пошук вразливостей по шаблону проводиться статично. При статичному аналізі досліджується код програми без його запуску. Код програмного забезпечення (в більшості випадків початковий) порівнюється з сигнатурами з бази методом побайтового порівняння або за допомогою складнішого алгоритму. При виявленні подібності, повідомляється про знайдену вразливість. Іноді сигнатура доповнюється деяким набором евристичних правил.

Проте, зазначені вище два підходи володіють рядом недоліків: перший важко піддається автоматизації, а другий припускає доступність вихідного коду програми. Тому в своїх дослідженнях ми вибрали третій підхід – fuzzing.

Останнім часом набуває популярності різновид методу тестування «сірого ящика», яке отримало назву фаззінг (fuzzing). Термін не так давно почали вживати і тому він є далеко не у всіх словниках. Найближче значення терміну «аналіз граничних значень» – визначення доступних діапазонів вхідних значень програми і тестування значень, які виходять за цей діапазон або знаходяться на межі. Фаззінг відрізняється тим, що не обмежує свою увагу на граничних значеннях, але так само займається підготовкою вхідних даних спеціального виду.

Список джерел:

1. https://www.owasp.org/index.php/Main_Page
2. Грег Хогланда и Гари Мак-Гроу. Взлом программного обеспечения.
3. http://www.itsec.ru/articles2/control/audit_progr_koda_treb_bezopasn
4. Fuzzing: исследование уязвимостей методом грубой силы. Саттон М., Грин А., Амини П.