УДК 004.056

НАЙПОПУЛЯРНІШІ АТАКИ НА ВЕБ-ДОДАТКИ ТА МЕТОДИ ПРОТИДІЇ ЇМ

Б. О. Стражник[1,a], С. А. Смирнов[1]

[1] Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського", НН Фізико-технічний інститут

Анотація

Ця робота була створена для опису найпопулярніших атак на веб-додатки та методів захисту від них на прикладі php фреймворку Laravel

Ключові слова: SQL Injection, XSS, API, XXE, php, Laravel, XML

Вступ

В час технологій веб-додатки стали невід'ємною частиною нашого життя. Зі зростанням кількості веб-додатків та складності їх функіоналу зросла й кількість та небезпека їх атак.

1. Найпоширеніші атаки на веб-додатки

SQL Injection

SQL-ін'єкції є однією з найбільш поширених атак на веб-додатки. SQL-ін'єкції дозволяють зловмисникам виконувати шкідливий код на сервері шляхом введення SQL-запитів у форми введення користувацьких даних, які потім відправляються на сервер для опрацювання запиту.

Вектори даної атаки можуть різнитися в залежності від типу конкретного веб-додатку та його вразливостей. Часто зловмисники використовують форми введення, такі як поля введення імені користувача або пароля, щоб вставити шкідливий SQL-запит, який відправляється на сервер бази даних. Ще частим вектором є параметри URL або заголовки HTTP запитів.

У 2021 році SQL-ін'єкції було віднесено до категорії Injection яка посіла 3 місце за поширеністю атак на веб-додатки. Загальна кількість зафіксованих випадків складала близько 275 тисячам. Статистика взята з джерела [1]

Кросс-сайт скриптинг

Кросс-сайт скриптинг (XSS) є атакою, при якій зловмисник впроваджує шкідливий скрипт у веб-сторінку, що виконується на стороні користувачів.

Вектор XSS атаки зазвичай включає введення шкідливого JavaScript-коду в поля введення веб-додатку, які потім зберігаються на сервері і відображаються на сторінках, до яких намагаються отримати доступ користувачі веб ресурсу. Зловмисники можуть використовувати різні методи для впровадження шкідливого коду включаючи вище зазначений метод, впровадження шкідливого скрипту в параметри URL або, найбільш розповсюджений метод, відправку шкідливих повідомлень на сайт через форму зворотнього зв'язку.

У 2021 році XSS атаки було віднесено до категорії Injection яка посіла 3 місце за поширеністю атак на веб-додатки. Загальна кількість зафіксованих випадків складала близько 275 тисячам. Статистика взята з джерела [1]

Атаки на сесії та кукі

Атаки на сесії та кукі є найбільш поширеними атаками на веб-додатки. Зловмисники можуть викрасти дані сесії або кукі, щоб отримати доступ до системи без необхідності, знання валідних логіну та паролю користувача.

Найпопулярнішим вектором атаки на сесії та кукі є MITM. MITM (Man-in-the-middle) – це атака на мережевий зв'язок, при якій зловмисник перехоплює трафік між двома вузлами спілкування та може змінювати і підміняти дані, що передаються.

При використанні MITM для атаки на сесії та кукі зловмисник перехоплює трафік між користувачем та сервером. Коли користувач надсилає запит на сервер для автентифікації, зловмисник перехоплює запит і захоплює сесійні дані та кукі, які використовуються для автентифікації користувача. Потім зловмисник може використовувати ці дані для виконання дій від імені користувача, не потребуючи повторної автентифікації.

Для захисту від MITM атак, необхідно використовувати захищене з'єднання з сервером за допомогою протоколу HTTPS, даний протокол забезпечує конфіденційність та цілісність даних шляхом шифрування трафіку. Таким чином все що може отримати зловмисник це зашифровані дані.

[a]bogstr-ipt23@lll.kpi.ua

У 2021 році атаки на сесії та куки були віднесені до категорії Insecure Design яка посіла 4 місце в статистиці найпоширеніших атак. Ці атаки складали частину від 262 тисяч зафіксованих випадків. Статистика взята з джерела [1]

Атаки на API

Зі зростанням популярності веб-додатків та мікросервісної архітектури, різні підрозділи яких фукціонують між собою за допомогою API, стали популярними атаки на API, які можуть призвести до витоку конфіденційної інформації або обходу механізмів безпеки.

Вектор атаки на API залежить від конкретного сервісу та сценарію використання, проте атаки на API авторизації та аутентифікації зазвичай вважаються найпоширенішими векторами. Тому важливо забезпечити достатній рівень безпеки при реалізації механізмів авторизації та аутентифікації, таких як використання сильних паролів, багатофакторна аутентифікація або токенів з обмеженим терміном дії.

Атаки на серверні компонети

Останніми роками зловмисники стали все частіше спрямовувати свої атаки на серверні компоненти веб-додатків. Наприклад атаки на вразливості у веб-серверах, вразливості конфігурації серверу, вразливості у хмарних сервісах тощо.

Вектори атаки на серверні компоненти відрізняються в залежності від конкретного сервера та використовуваних технологій. Проте, атаки на мережеву інфраструктуру, вразливості в базах даних зазвичай є найпоширенішими векторами даних атак. Тому важливо забезпечувати безпеку серверів, встановлюючи оновлення всіх компонентів, використовуючи засоби моніторингу та захисту, такі як брендмауєри, системи виявлення вторгнень та системи аудиту.

У 2021 році ці атаки було віднесено до категорії Vulnerable and Outdated Components, і загальна кількість зафіксованих випадків становила близько 31 тисячі. Статистика взята з джерела [1]

XXE атаки

XML External Entity атаки -– це вразливості веб-додатків, які дозволяють зловмисниками отримувати віддалений доступ до файлової системи та інших ресурсів сервера. Атака полягає в експлуатації вразливості парсера XML, який дозволяє посилання на зовнішні сутності в XML-документі.

Один з найпоширеніших векторів використання XXE-атак -– це атака на систему обробки XML-даних, де зловмисник використовує впроваджений код у XML-документі, який викликає читання чи запис файлів на сервері або передачу конфіденційної інформації через мережу.

У 2021 році XXE-атаки було віднесено до категорії Security Misconfiguration яка займала 5 місце, загальна кількість зафіксованих випадків становила близько 209 тисяч. Статистика взята з джерела [1]

2. Засоби захисту на рівні веб-додатку на прикладі php фреймворку Laravel

Приклади php коду було взято з джерела [2]

2.1. SQL Injection

- Використання підготовлених запитів з параметрами

```
$id = Auth()->user()->id;
$query = 'SELECT␣*␣FROM␣Users␣WHERE␣id␣=
    ␣?';
$user = DB::select($query, [\$id]);
```

- Використання Eloquent ORM
  В данному прикладі було використано Eloquent ORM через те, що ця техніка запитів до бази даних є встроєною в Laravel фреймвор, в інших фреймворках також бажано використовувати ORM системи, наприклад Prisma.
  Чим же ORM системи допомогають в запобіганні SQL-ін'єкцій, вони автоматично використовують параметризовані запити, чим підвищують developer experience та захист від даного типу атак.

```
$id = Auth()->user()->id;
$user = User::where('id', \$id);
```

2.2. XSS атаки

- Використовуйте функції, що екранують спецсимволи при виведенні даних на сторінку, наприклад, htmlspecialchars()
  Приклад використання данної функції в .blade файлах

```
{!!  htmlspecialchars(\$post->code) !!}
```

  Або використання вбудованої дерективи в .blade файли яка автоматично конвертує спеціальні символи в рядок

```
{{ \$post->code }}
```

  Приклад такої функції в фрейморці NextJS

```
import { escape } from 'html-escaper'
const myString = "<script>alert('Hello␣
    World')</script>"
const escapedString = escape(myString)
```

  Приклад було взято з джерела [3]

- Виконуйте валідацію всіх даних які надходять від користувачів

```
$request->validate([
  'name'=>['required', 'regex:/^[A-Za-z0
      -9\s]+$/'],
  'email'=>['required', 'email']
]);
```

В даному прикладі ми використовуємо вбудовані механізми валідації вхідних даних в Laravel, в інших фреймворках ви можете використати бібліотеки по типу formik

2.3. Атаки на сесії та кукі

- Завжди використовуйте шифрування сесій та кукі, в фреймворці Laravel це робиться в файлі **config/session.php**, потрібно поставити true, навпроти ключа encrypt
- Використовуйте HttpOnly та secure атрибути для кукі
  В фреймворці ларавель це робиться в файлі **config/session.php**
  В інших фреймворках, наприклад Next JS це можливо зробити при створенні кукі

```
const cookieStr = cookie.serialize('my-
    cookie-name', value, {
  maxAge,
  expires: new Date(Date.now() + maxAge
      * 1000),
  httpOnly: true,
  secure: true,
});
```

2.4. Атаки API

- Використовуйте токени доступу та ліміти на запити

```
Route::middleware(['throttle:10,1', '
    auth'])->get('/api', function () {
    return 'Hello␣World';
});
```

- Використовуйте сторонні сервіси для захисту від DDoS атак, наприклад такі як CloudFlare
- Використовуйте HTTPS протокол для захисту даних

2.5. XXE атаки

- Використовуйте XML-парсери з блокуванням використання сторонніх сутностей

Висновки

У підсумку можна констатувати, що атаки на веб-додатки залишаються одним із найбільших ризиків для інформаційної безпеки, так як у разі їх успіху зловмисники можуть отримати доступ до серверу, конфіденційної інформації користувачів, такої як дані платіжної системи, логіни, паролі, електронні скриньки та багато іншої, яку в подальшому можуть використовувати для отримання вигоди. Однак ви можете значно знизити ризики успішних атак на рівні веб-додатку, використовуючи рекомендації по протидії атакам, які були описані в даній роботі, а саме завжди валідувати дані які надходять від користувачів, використовувати лише захищені канали зв'язку між користувачами та сервером та інші.

Перелік використаних джерел

1. OWASP Top 10 / OWASP. — 2021. — URL: https://owasp.org/Top10/.
2. Laravel Documentation / Laravel. — 2023. — URL: https://laravel.com/docs/10.x.
3. html-escaper / WebReflection. — 2023. — URL: https://github.com/WebReflection/html-escaper.