1. **Описание.**

Juice shop – это веб-приложение, построенное на основе node.js.

Является “интернет-магазином”. Является объектом для статистического анализа кода, который был взят с GitHub разработчика.

2.**Сканируем через semgrep в CLI. Результаты сканирования**















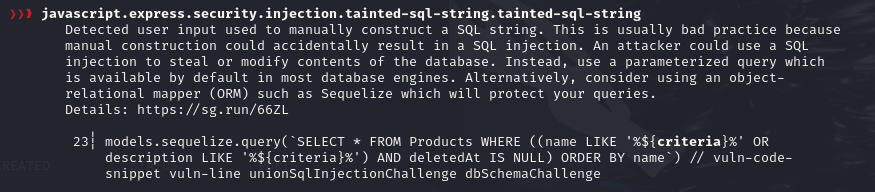




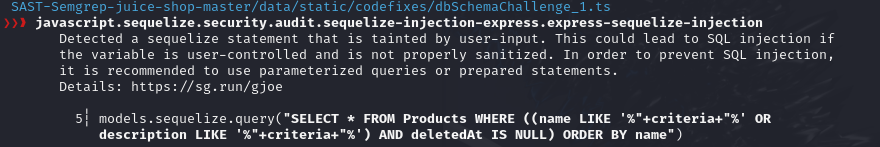
**3. Уязвимости из OWASP Top-10, обнаруженные в результате статического анализа, — минимум пять штук**

1. SQL инъекции

1.1

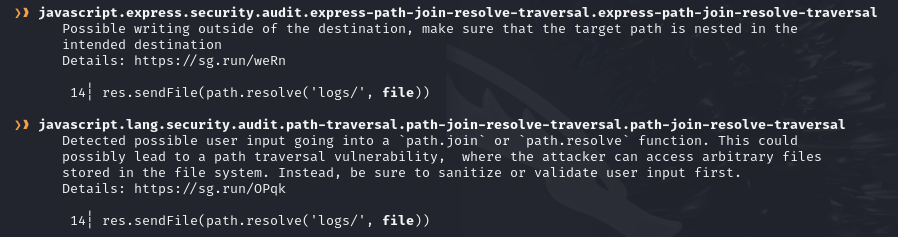


1.2



1. **Broken Access Control**.

2.1 Уязвимость path traversal



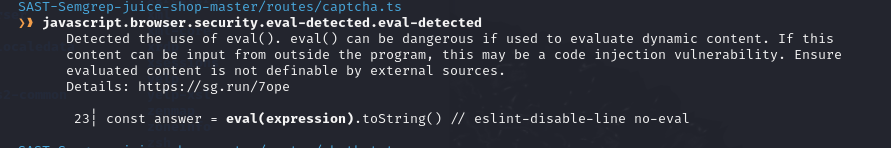
2.2 Res send file

Основная проблема с res.sendFile() связана с **атаками обхода каталогов (Directory Traversal)**, когда для определения пути к файлу используется пользовательский ввод.



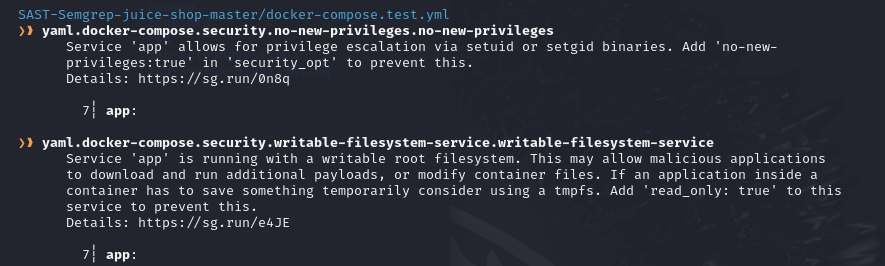
2.3

Уязвимость eval-detected — это правило в инструменте Semgrep, которое предупреждает об опасности использования функции eval() для оценки динамического контента. Если этот контент может быть введён извне программы, это может указывать на уязвимость инъекции кода



2.4

Доступна для чтения файловая система



1. Ошибки криптографии

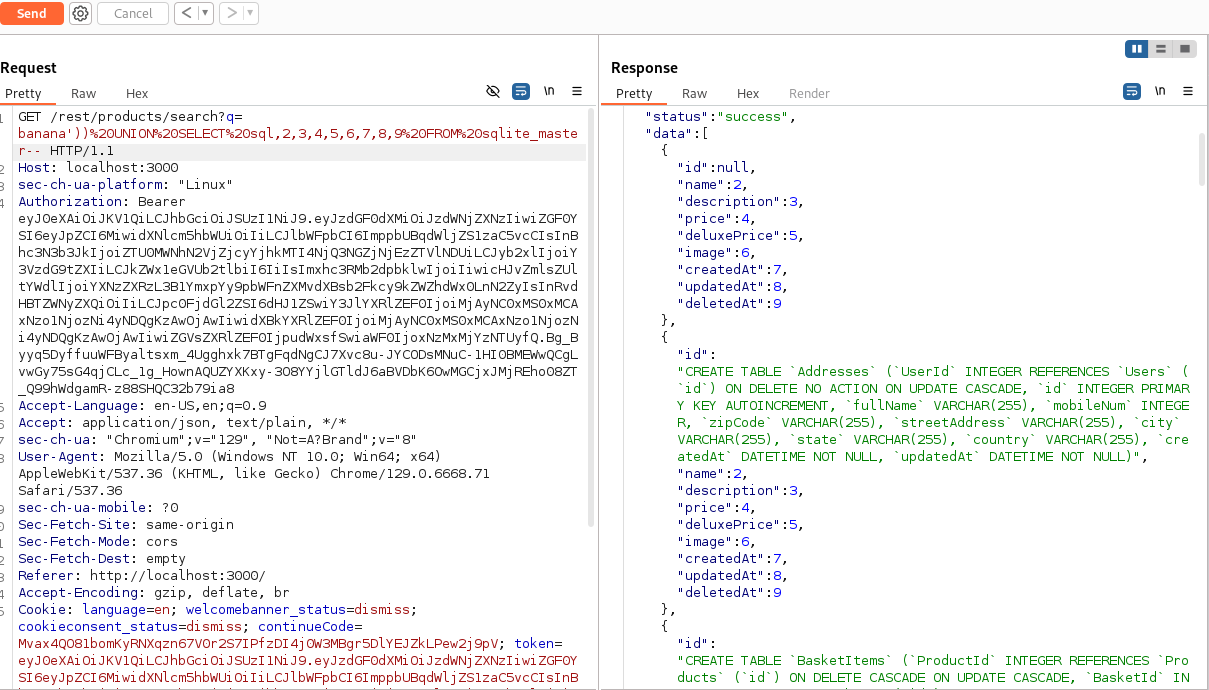
3.1 detected-jwt-



**4. Демонстрация эксплуатации трёх уязвимостей из OWASP Top-10 — скриншоты эксплуатации, проведённой с помощью инструмента Burp Suite. Выбор этих уязвимостей остаётся на ваше усмотрение.**

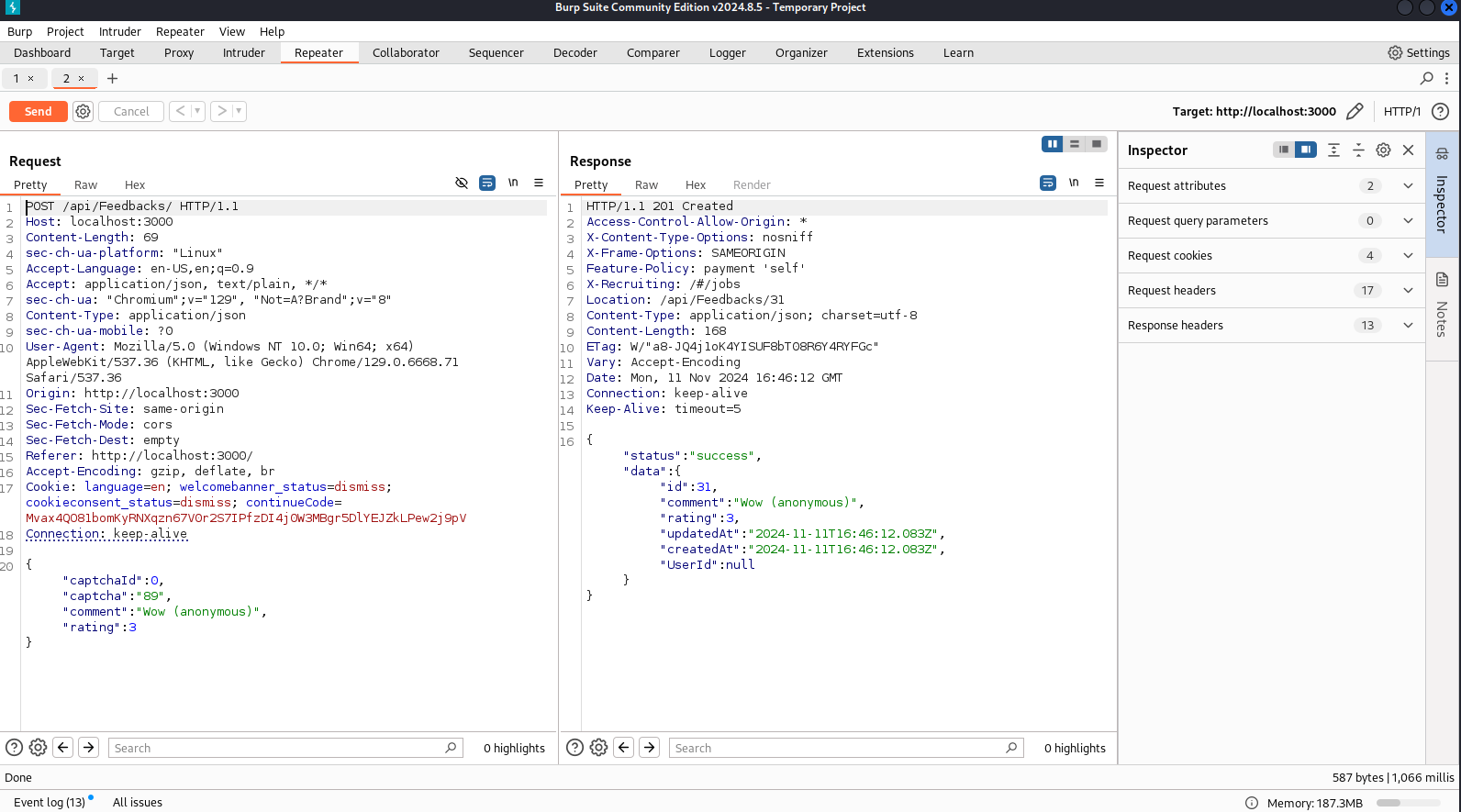
1. **SQL инъекция. Раскрытие схемы базы данных.**

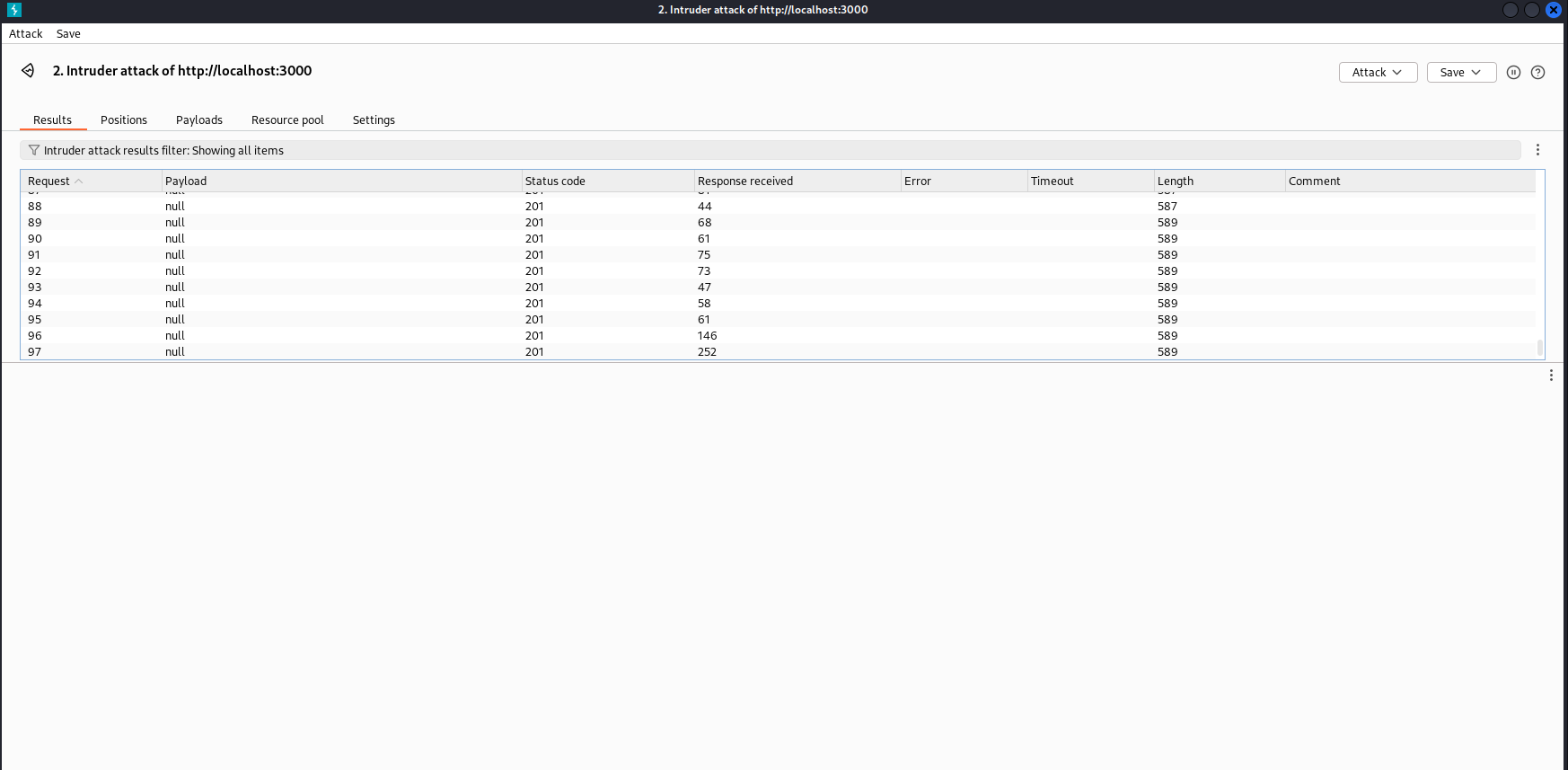
Из документации sqlite узнаем, что в системе БД sqlite существует мастер таблица, где указаны все существующие таблицы в системе БД. С помощью запроса union select методом перебора находим количество столбцов – их 9.

****

1. **Broken Authentication and Session Management**

В приложении нет ограничений по количеству post запросов к серверу. Значит мы можем нагрузить его бесконечными запросами/публикациями в разделе feedback или решениями капчи.

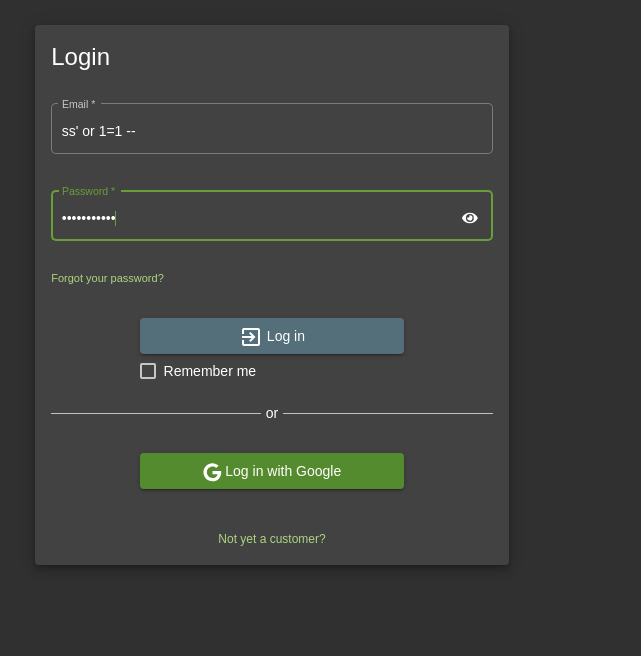


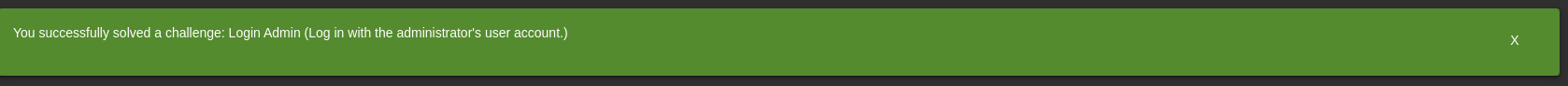


1. **Sql инъекция**.

Входим в админскую учетную запись с помощью инъекции в поле ввода.

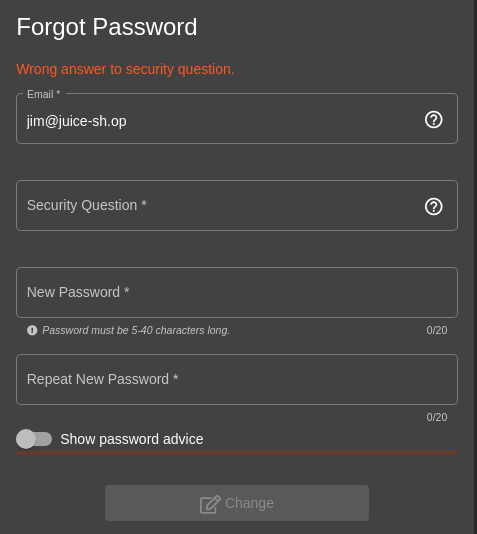
Сервер воспринимает наш запрос 1=1 корректным и пропускает.



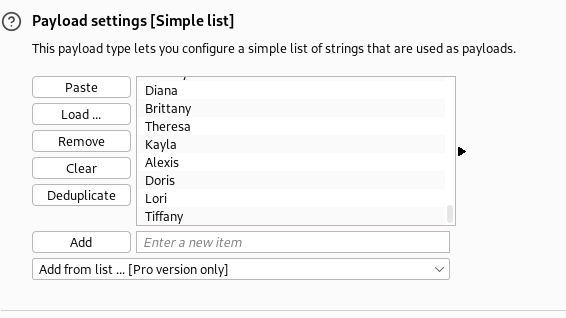


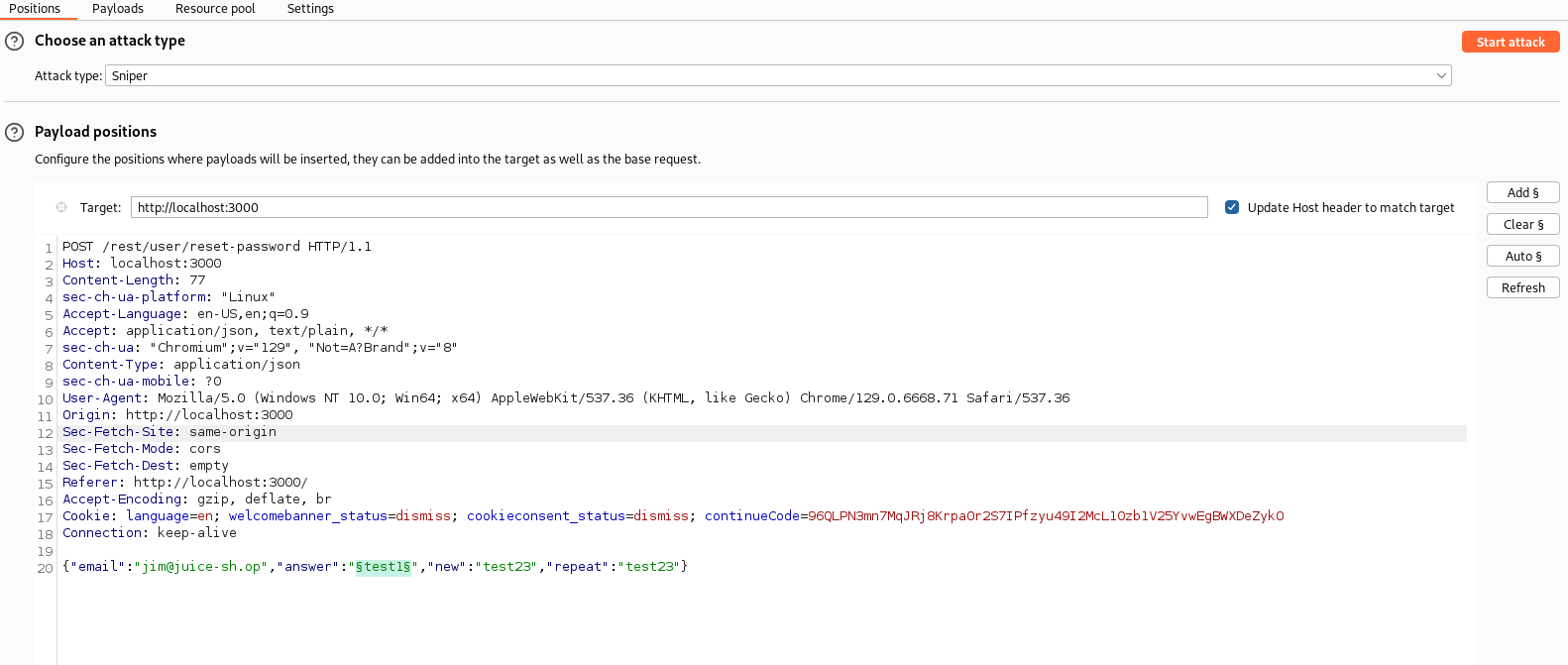
1. **Broken Access Control**

Заполняем форму для нового пароля пользователя Jim. Находим, что мы ввели неправильно секретный ответ на вопрос.

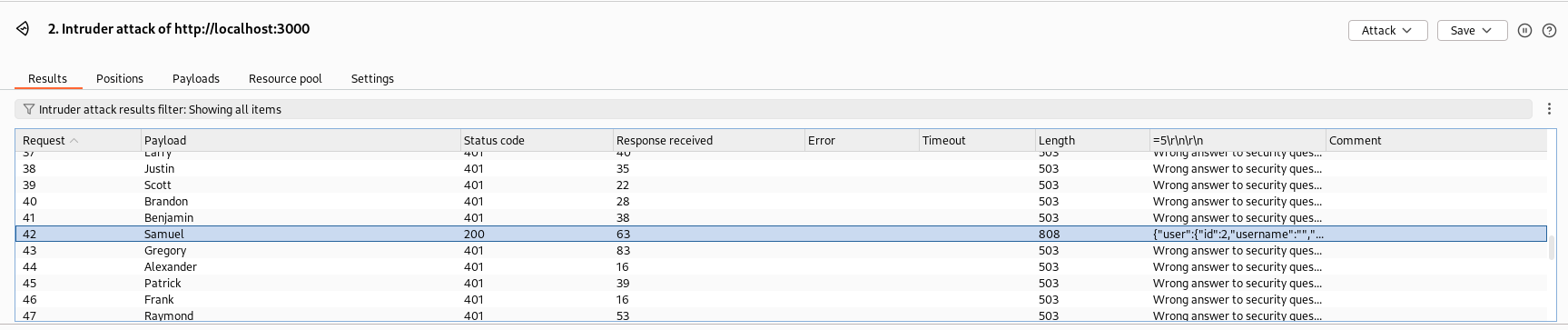


Находим POST метод в Proxy HTTP history в burpsuite и переносим его в intruder. Также находим топ самых популярных имен в гугле. И вставляем в Payload Intruder.

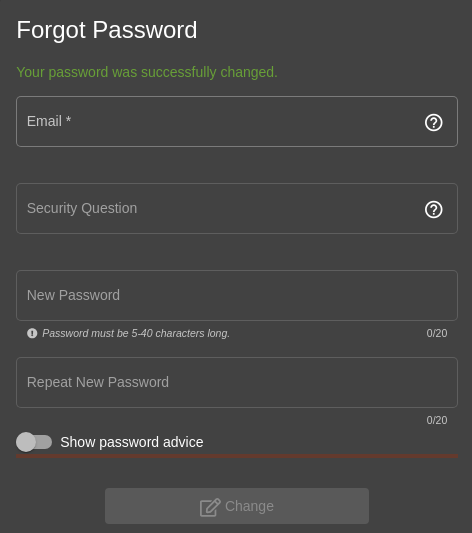




В результатах брутфорса видим, что одно имя дает не такой результат, как все остальные.



Успешно меняем пароль



1. Рекомендации по устранению продемонстрированных уязвимостей.

А. Постоянное статическое и динамическое сканирование приложения.

Б. Тестирование на наличие уязвимостей

В. Ограничение привилегий (Принцип наименьших привилегий)

Г. Использование параметризованных запросов