Noda-Group

OWASP Juice Shop

Внутренний тест на проникновение
Отчет о результатах

OWASP Juice Shop
Июль 16, 2023
Version 1.0

Оглавление

Заявление о конфиденциальности

- Настоящий отчет о проведении аудита уязвимостей веб-приложения создан в строгом соответствии с принципами конфиденциальности и информационной безопасности. Документ содержит информацию, относящуюся к безопасности и защите информационных систем организации.
- Данный документ и содержащаяся в нем информация предназначены исключительно для внутреннего использования компетентными сотрудниками организации, участвующими в процессах обеспечения безопасности информационных систем.
- Распространение, копирование, передача или любое иное использование информации из данного отчета, без явного письменного согласия ответственного лица организации, строго запрещены. Несоблюдение этого требования может повлечь за собой дисциплинарные, административные, гражданско-правовые или уголовные последствия, в зависимости от тяжести нарушения и причиненного ущерба.
- Все лица, имеющие доступ к данному отчету, обязаны обеспечивать конфиденциальность информации, содержащейся в нем, и не использовать ее в иных целях, кроме как для обеспечения безопасности веб-приложения. Любое отклонение от этого требования требует получения явного письменного разрешения со стороны уполномоченного лица организации.
- Подчеркивается строгая ответственность каждого, кто имеет доступ к этому отчету, за обеспечение конфиденциальности и сохранность содержащейся в нем информации.

Контакты для участия

	Inlanefreight Contacts	
Primary Contact	**Title**	**Primary Contact Email**
Pavlov Dmitry	OWASP SEO	pavlov@owasp.com

Ввод в отчет
OWASP Juice Shop. («juice-shopt» в настоящем документе) заключила контракт с NodaGrup на проведение теста на проникновение в сеть
Внутренняя сеть juice-shopt для выявления слабых мест в безопасности, определения влияния на juice-shopt,
четко и воспроизводимо задокументируйте все результаты и предоставьте рекомендации по исправлению положения.
https://demo.owasp-juice.shop/#/

Подход

Noda Group провела тестирование по методу «черного ящика»

с 10 июля 2023 г. по 15 июля 2023 г.
При проведении данного исследования был использован метод "Черного ящика" в тестировании веб-приложения OWASP Juice Shop. Данный метод был выбран компанией OWASP Juice Shop , не требующего предварительного знания о внутренней структуре или функциональности тестируемого веб-приложения.

Целью данного тестирования являлось исследование и выявление возможных уязвимостей безопасности внутри приложения, которые могут быть эксплуатированы злоумышленниками.

На протяжении всего процесса тестирования основное внимание уделялось исследованию онлайн версии веб-приложения OWASP Juice Shop и анализу инфраструктуры его локальной сети. Такой комплексный подход к тестированию позволил нам оценить глобальные и локальные аспекты безопасности системы, включая процессы аутентификации, авторизации, а также возможности перехвата и манипулирования данными. Особое внимание было уделено следующим уязвимостям : XSS, SQL injection, Broken authorization/broken authentication, Informational disclosure.

Процесс тестирования был направлен на обнаружение угроз безопасности, которые могут иметь место как в процессе взаимодействия пользователя с приложением, так и на уровне сервера. Каждая выявленная уязвимость была подробно изучена и документирована для дальнейшего анализа и разработки мер по ее устранению. Также в этом документе присудствет решение найденых уязвимостей в формате short-term и long-term.

Объем

Области этой оценки были : внутренняя сеть, и онлайн версия вэб-приложения . OWASP Juice Shop

Host/URL/IP Address	Description
https://demo.owasp-juice.shop/#/	OWASP online web-aplication

Обзор оценки и рекомендации к устранению Уязвимостей

В ходе проведения тестирования нашей командой "Noda-Group", были выявлены ряд уязвимостей в веб-приложении "OWASP Juice Shop", которые включают в себя проблемы с XSS (Cross-site Scripting), SQL Injection, Broken Authentication/Broken Authorization, Data Base Scheme Injection, HTML Injection, обнаружение чувствительных данных, и информационное раскрытие.
XSS и HTML Injection требуют внедрения строгих мер валидации и санитизации вводимых данных, а также применения механизмов контроля доступа к данным (Content Security Policy). SQL Injection и Data Base Scheme Injection требуют также применения санитизации входных данных и использования параметризованных запросов или ORM (Object-Relational Mapping) систем.
Broken Authentication/Broken Authorization и информационное раскрытие указывают на необходимость пересмотра политик аутентификации и авторизации, улучшения управления сессиями и обеспечения конфиденциальности данных пользователей. Чувствительные данные должны быть должным образом защищены и шифрованы во время хранения и передачи, чтобы предотвратить несанкционированный доступ.

Ввод и оценивание теста на проникновение в ВЭБ-Приложение

NodaGroup подошла к вопросу тестирования максимально реалистично, а именно были проведены тесты в формате «Блэк Бокс» без выданных айпи адресов , специальных аккаутов или доступов. Тесты были проведены таким образом и с таким уклоном в тестировании, что бы проверить какой уровень вреда теоретически может нанести злоумышленник, в такой же ситуации. Единственное что было предоставлено это доменное имя цели тестирования

Резюмированные результатов проделанной работы.

В ходе проделанных тестов было выявлено 8 уязвимостей . Уровень уязвимостей от «Критического» до «Высокого» соответственно в критическом состоянии находится Вэб-Приложение и под большой угрозой вся персональная информация пользователей и администраторов.

Finding #	VAT priority	VRT category	Finding Name
1	P1	Server-Side Injection	SQL injection (доступний ввод SQL вредосносного кода)
2	p1	Broken Authentication	Using Default Credentials/pass (использование дефолтного пароля для администратора)
3	p1	Insecure OS/Firmware	manipulation No sql injection/Command Injection (получили всю базу данных пользователей и Администраторов)
4	p1	Server Security Misconfiguration	users credentials (Получили доступ к логам всех пользователей за период)
5	p1	**Sensitive Data Exposure**	Получили доступ к файлу с бєкапами сервера (package.json.bak)
6	p1	Sensitive Data Exposure	Disclosure of Secrets (Путем сканирования дерикторий были найдены Конфиденциальные документы)
7	p1	Broken Authentication and Session Management	Unauthorized Privilege Assignment (была создана учетноя запись и наделена правми администратора)

Пошаговое руководство по компрометации вэб – приложения OWASP Juice Shop.

В ходе проверки OWASP Juice Shop компании Noda-Group удалось скомпрометировать ресурс (Juice Shop), получить чувствительные данные самого вэб приложения в виде истории бэкапов, полный перечень продуктов и товаров которые находится на вэб-приложении в частности скрытую информацию, получили доступ к учетным данным администраторов также доступ к информации и персональным данным клиентов \ пользователей. Также получили полный доступ к администрированию ресурса с возможностью Выгрузки запрещённых файлов (в частности те которые не отображаться по FTP доступу к серверу и не возможно было просмотреть эти файлы с клиентской части.) и Загрузки файлов неразрешённых типов и вредоносных скриптов.

Подробное прохождение.

Компании «Noda-Grup» удалось скомпрометировать «OWASP Juice Shop» благодаря следящим действиям. (Приведен перечень пошаговых действий)

1) Была произведена предварительная подготовка, в частности отсмотрен и протестирован базовый функционал вэб-приложения, другими словами были эмитированы действия обычного пользователя (https://demo.owasp-juice.shop/#/)

2) Просканирован домен через «nmap» и выявлен айпи адрес и открытые порты «81.169.145.156»

3) Запущено сканирование директорий вэ-приложения через инструмент «dirb» было найдено некоторые директории , тестер будет использовать (ftp, support, log, administrtions and other…)

4) Запущено сканирование файлов вэб приложения через инструмент «FFUF» были найдены определённые файлы, тестер будет использовать (package.json, logfile.bk,)

5) Была произведена попытка классической «SQL-injection» в окно ввода учетной записи и сразу же получен доступ администратора. «https://demo.owasp-juice.shop/login» и стразу был получен доступ к «https://demo.owasp-juice.shop/profile»

6) Был произведен сброс пароля Администратора и тем самым проверенная функция «Broken Access control /Authentication Bypass»

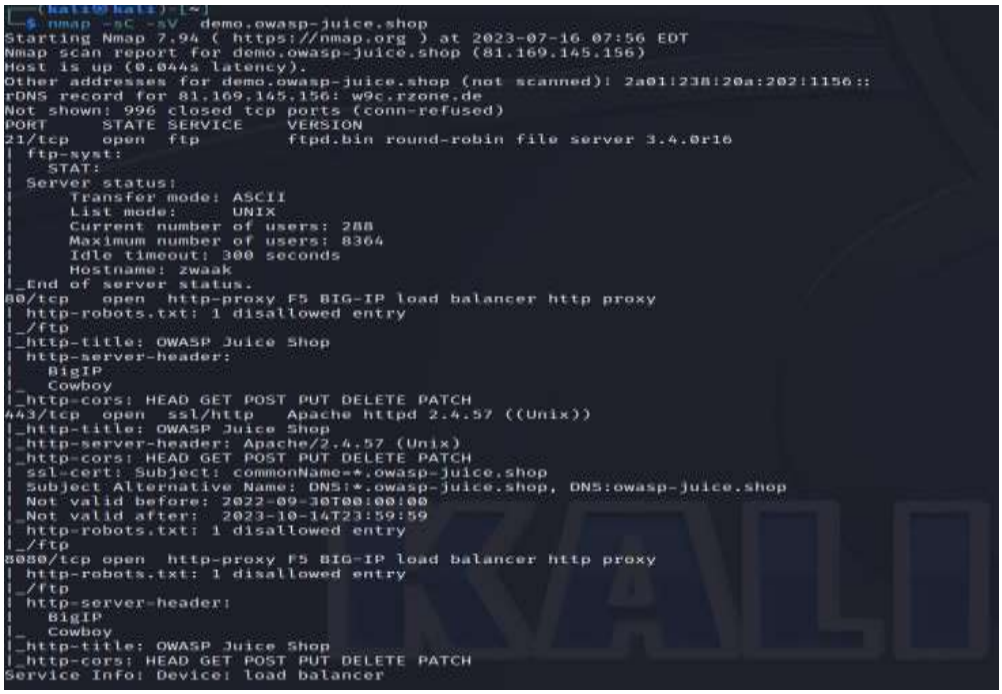
7) Получен доступ к файлу «package.json» при помаши инструмента «FFUF» и уязвимости «bypass access restrictions», и это даёт нам дополнительную информацию о архитектуре вэб-приложения и его уязвимостях.

8) Была виявлена узвимость в базе данных отзывов о товаре и проэксплуатирована путём добавлений определённого сообшения от имени всех пользователей. Это действие быо выполнено через инструмент «BurpSuite» и из-за присудствия уязвимоти «NoSQL Manipulation»

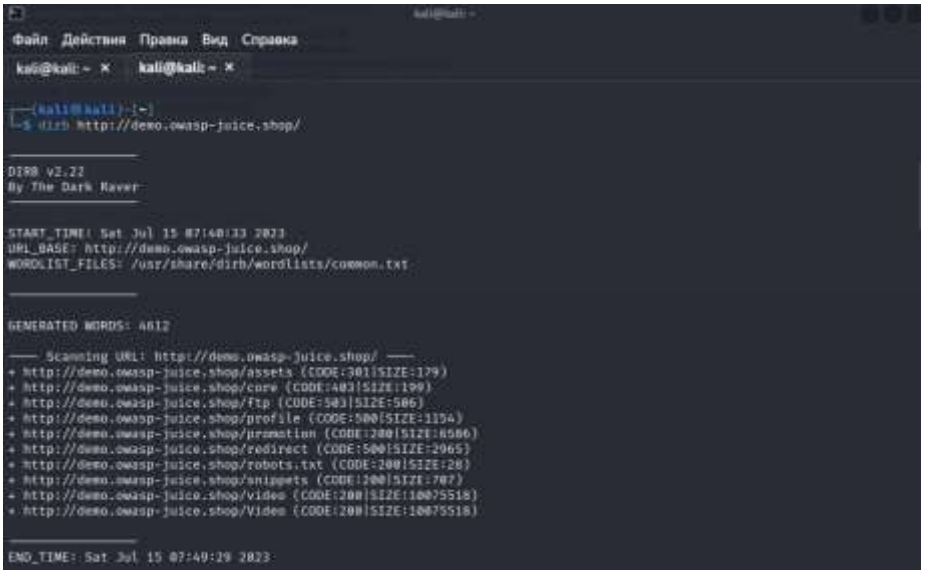
9) При помоши інструмента «BurpSuite» были найдены и использованы узвимости «bypass access restrictions» и «NoSQL-injection» получен доступ ко всем акаунтам пользовтаелей и администоров .

Пошаговое руководство по компрометации вэб-приложения

2)Сканирование домена на предмет открытых портов и айпи адреса



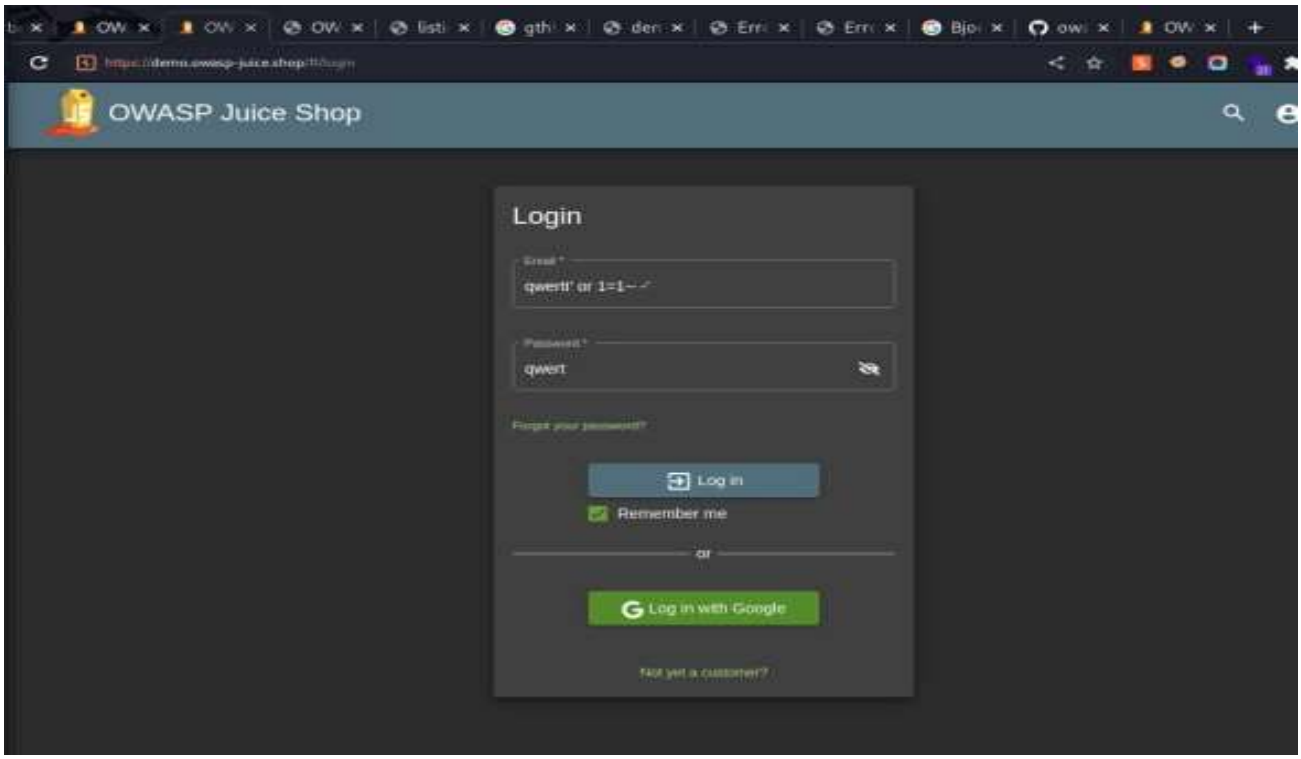
3)Сканируем дериктории при помоши интрумента «dirb» и получаем видимые дериктории



4)Переходим в инструмент «FFUF» и запускаем сканирование файлов доменного имени и получаем результат в виде найденных файлов и директорий.

```
root@kali: ~
Файл  Действия  Правка  Вид  Справка
┌──(root㉿kali)-[~]
└─# ffuf -w /usr/share/dirb/wordlists/common.txt -u https://demo.owasp-juice.
shop/FUZZ

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v2.0.0-dev
________________________________________________

 :: Method           : GET
 :: URL              : https://demo.owasp-juice.shop/FUZZ
 :: Wordlist         : FUZZ: /usr/share/dirb/wordlists/common.txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200,204,301,302,307,401,403,405,500
________________________________________________

:: Progress: [1/4614] :: Job [1/1] :: 0 req/sec :: Duration: [0:00:00] :: Err
:: Progress: [40/4614] :: Job [1/1] :: 0 req/sec :: Duration: [0:00:00] :: Er
:: Progress: [40/4614] :: Job [1/1] :: 0 req/sec :: Duration: [0:00:00] :: Er
[Status: 403, Size: 199, Words: 14, Lines: 8, Duration: 47ms]
    * FUZZ: .htaccess

:: Progress: [40/4614] :: Job [1/1] :: 0 req/sec :: Duration: [0:00:00] :: Er
[Status: 403, Size: 199, Words: 14, Lines: 8, Duration: 61ms]
    * FUZZ: .hta
```

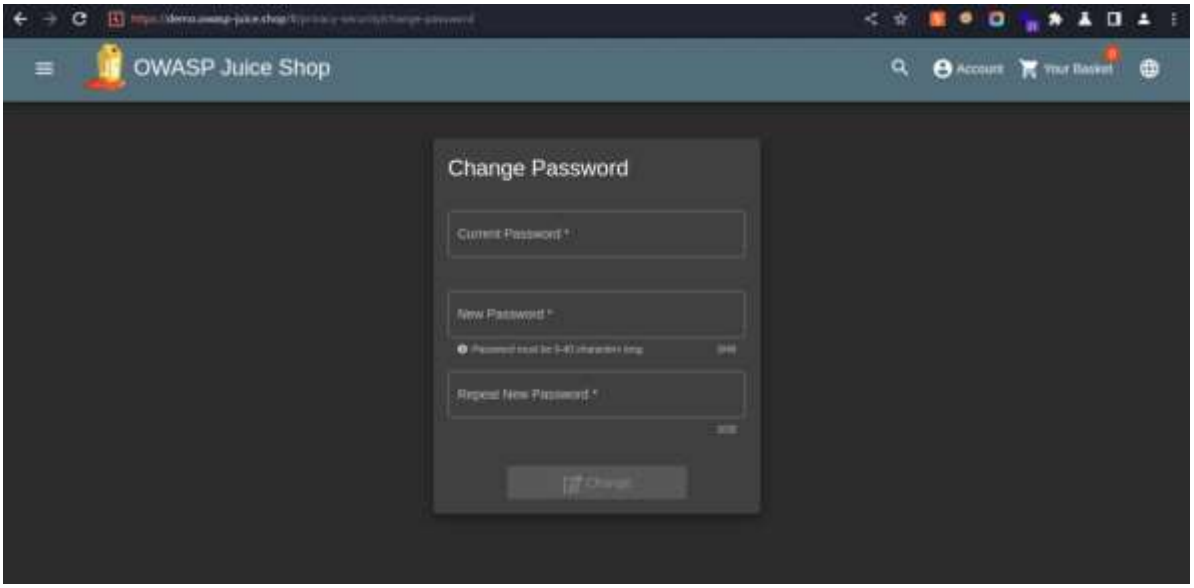
5)Параллельно с этим на нам необходимо перейти по адресу «**https://demo.owasp-juice.shop**» и войти в учетную запись. При этом обращаю внимание что помимо «SQL – Injaction» которая вызывается путем ввода слеующей комбинации « ' or 1=1-- -'»



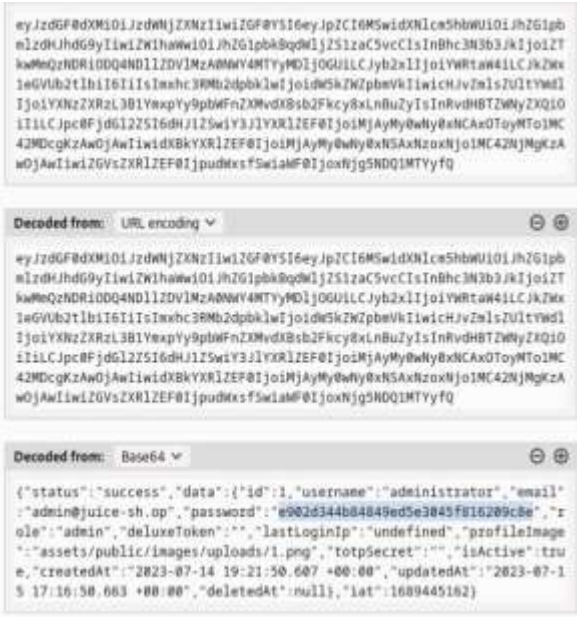
Наблюдается проблема , дефолтного пароля учётной записи «admin@juice-sh.op» пароль «admin123»

6)Первым делом надо войти в учетную запись «https://demo.owasp-juice.shop/#/login» потом перейти по ссылке «https://demo.owasp-juice.shop/#/privacy-security/change-password» и открыть инстркмент «BurpSutie» .Переходим на вкладку «proxy» далее «intercept in» и активируем (это статический процес перехвата запросов между клиентской частью и сервером, далее будем использовать термин «перехват запроса»

далее ,открываем «https://demo.owasp-juice.shop/#/privacy-security/change-password» вводим в графу старый пароль « любое значение» , а в две остальные поля вводим - новый пароль » Нажимаем кнопку подтвердить



Возвращаемся в «BurpSutie» отправляем запрос в репитер, в самом Репитере Выделяем значение в нвшем случае «matinam»



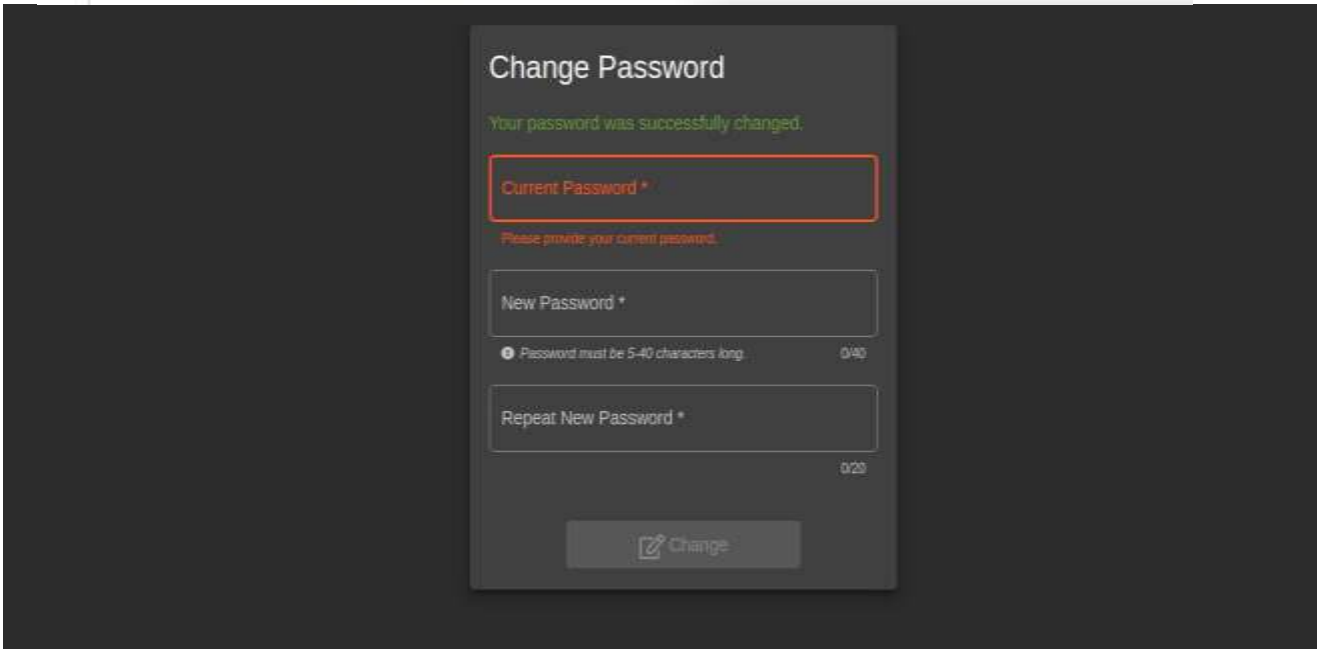
Переходим в сервис определние хеша (https://hashes.com/ru/tools/hash_identifier)и доовлем туда значение хеша. Добавив значение хеша, становится понятно что формат хеширования md5.

Переходим в «Repiter» и добавляем полученное значение в поле как на скриншоте. И отправляем запрос

```
GET /rest/user/change-password?current=matinam&new=qwerty&
repeat=qwerty HTTP/2
Host: demo.owasp-juice.shop
Cookie: welcomebanner_status=dismiss; cookieconsent_status=
```

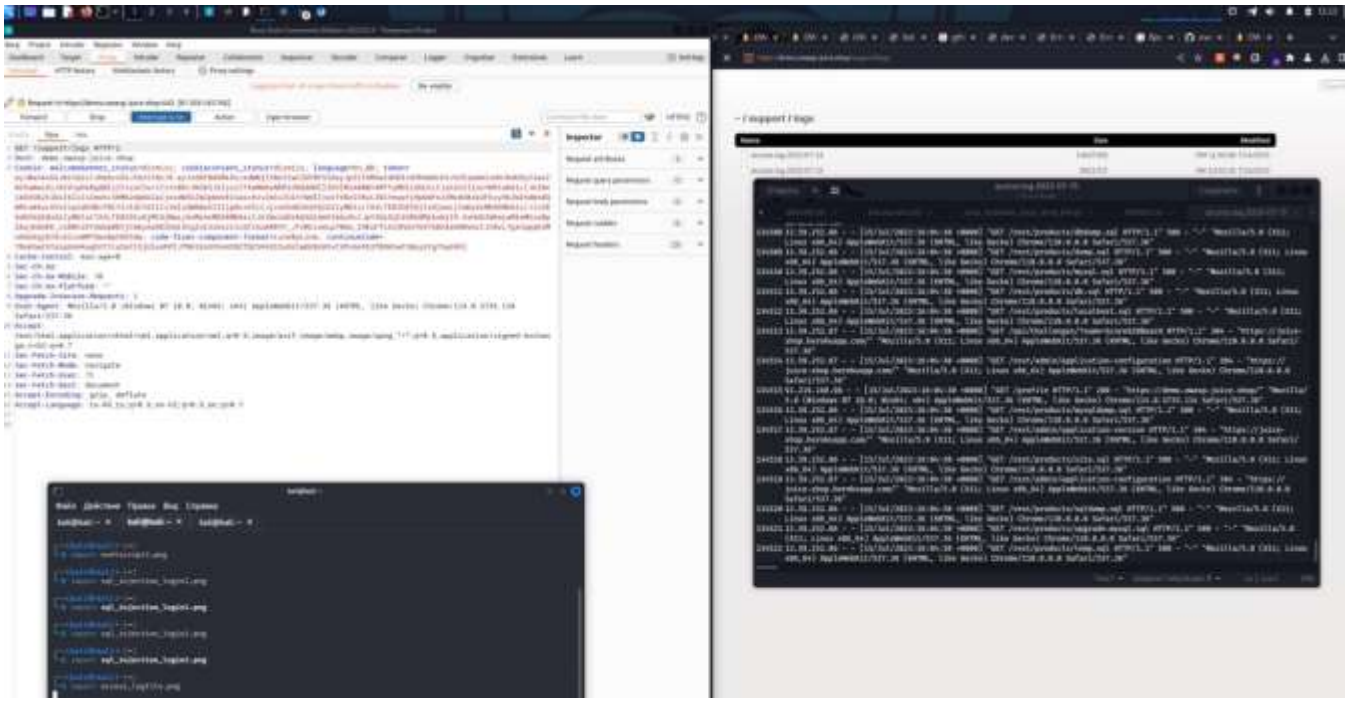
Все задача выполнена пароль изменен.


```
"user":{
  "id":1,
  "username":"",
  "email":"admin@juice-sh.op",
  "password":"d8578edf8458ce06fbc5bb76a58c5ca4",
  "role":"admin",
  "deluxeToken":"",
  "lastLoginIp":"undefined",
  "profileImage":
  "assets/public/images/uploads/defaultAdmin.png",
  "totpSecret":"",
  "isActive":true,
  "createdAt":"2023-07-15T21:27:02.536Z",
  "updatedAt":"2023-07-16T01:21:24.912Z",
  "deletedAt":null
```

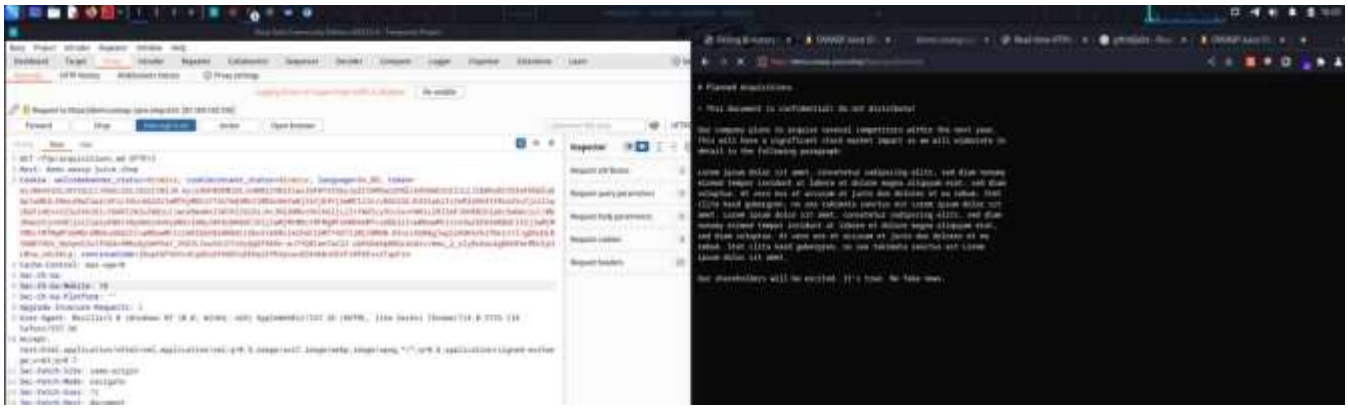
7)Запускаем сканет «FFUF» получаем перечен отсканированых дерикторий и файлов.

8)Переходим по ссылке «https://demo.owasp-juice.shop/support/log» и видим файл открываем его через дополнительную команду «https://demo.owasp-juice.shop/support/log/access.log.2023-07-14%25%30%30.md»



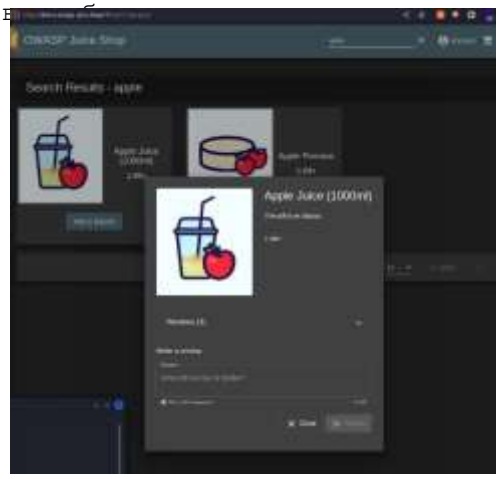
Получен файл с логами всех пользователей

9))Переходим по ссылке «https://demo.owasp-juice.shop/ftp/acquisitions.md» и видим файл с конфиденциальной информацией.

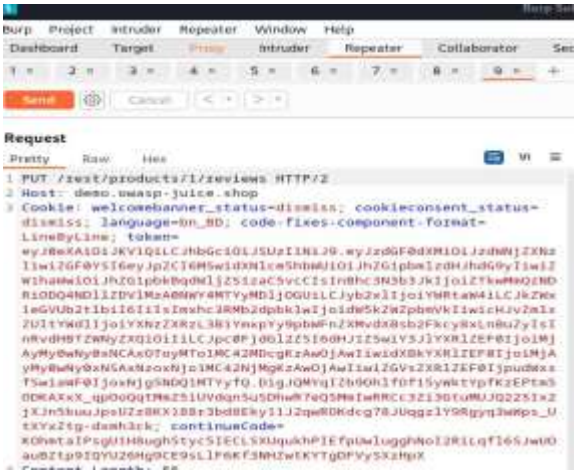


10)По точно такому же принципу переходим по ссылке «https://demo.owasp-juice.shop/ftp/package.json» полученной путем сканирования через «FFUF» и «dirb».
Сразу файл открыть не получится, добавляем к ссылке следующие значения «%25%30%30.md» и ссылка на выходе у нас получается следующая «https://demo.owasp-juice.shop/ftp/package.json%25%30%30.md»

11)Переходим по ссылке «https://demo.owasp-juice.shop/ » выполняем поиск «apple » и отставляем отзыв о товаре «DmitryPavlov_good_teacher» открываем инструмент «BurpSute» выполняем процесс перехвата запроса через «Proxy>Intercept» отправляем запрос в «Repiter», в самом запросе замещаем «"message":"dsadasdsdadas",«"author":"admin@juice-sh.op"» на «{"message":"DmitryPavlov_good_teacher","id":{"$ne":-1}}» и замещаем «POST /rest/products/1/reviews» на «PATCH /rest/products/reviews» отправляем запрос на серв



screenshot1

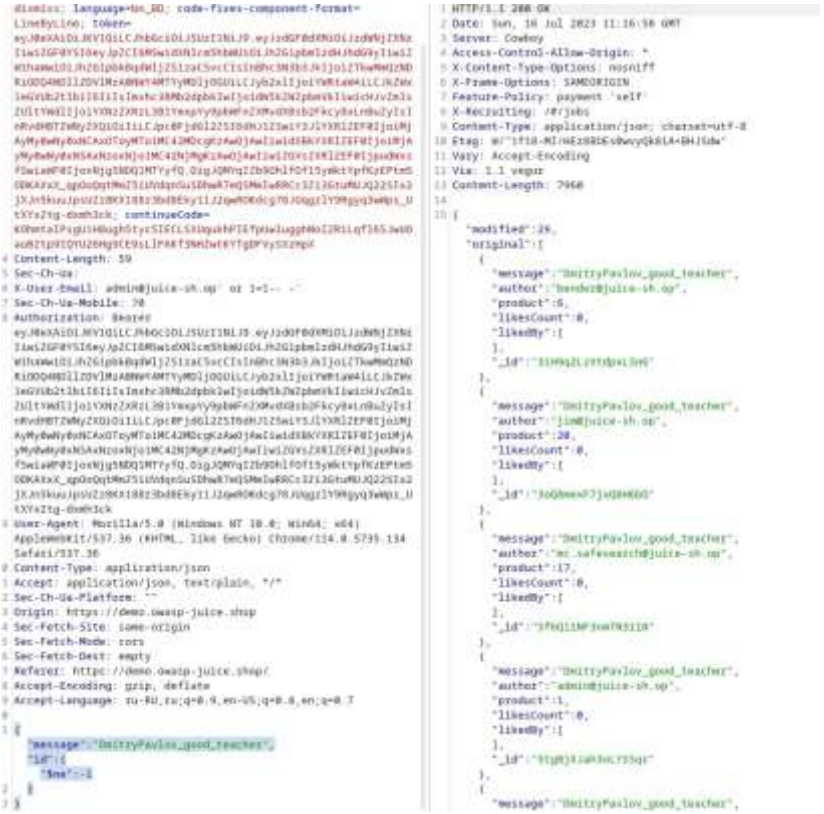


screenshot2

screenshot3

screenshot4

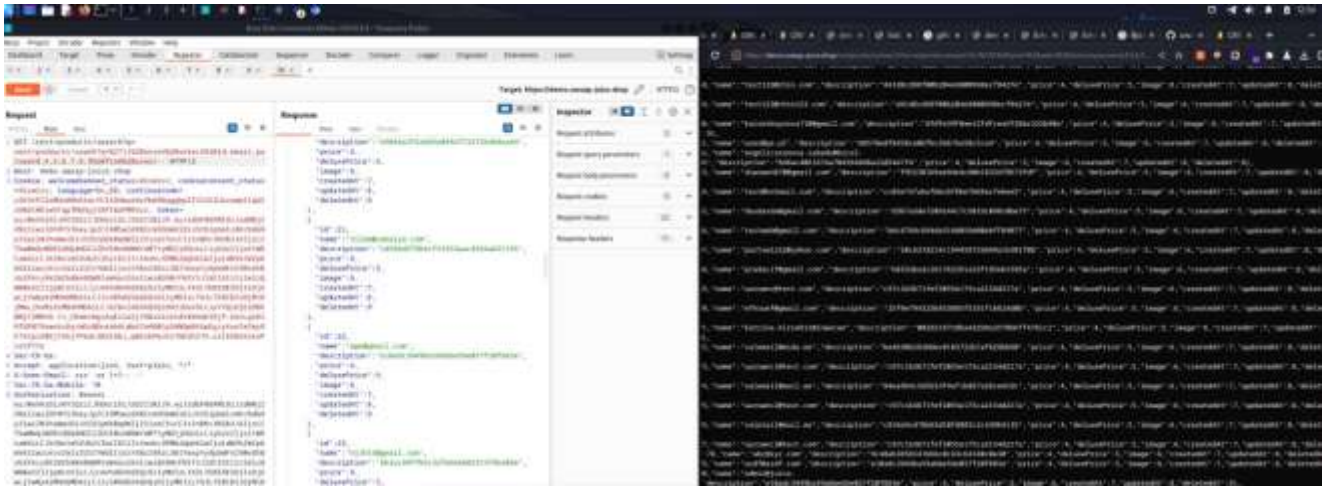
screenshot5



screenshot5

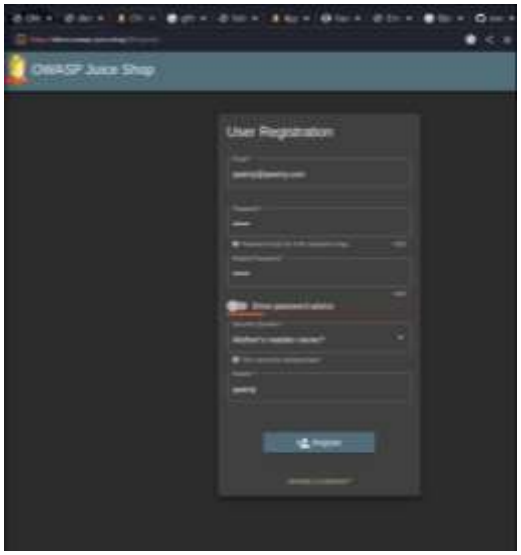
В окне вывода видим полученную информацию, свидетелствует о том что мы успешно проэксмплуатировалили уязвимость и отправили отзыв от всех пользователей.

12)Переходим по ссылке «https://demo.owasp-juice.shop» выполнеям команду поиска «apple» предварительно перехватив запрос через через «Proxy>Intercept» отправляем запрос в «Repiter», меняем значение :
«/rest/products/search?q=apple»
на следующее значение
«/rest/products/search?q=rest/products/searh?q=%27))%20union%20select%20id,email,password,4,5,6,7,8,9%20from%20users--»
Получаем логины и пароли всех пользователей.



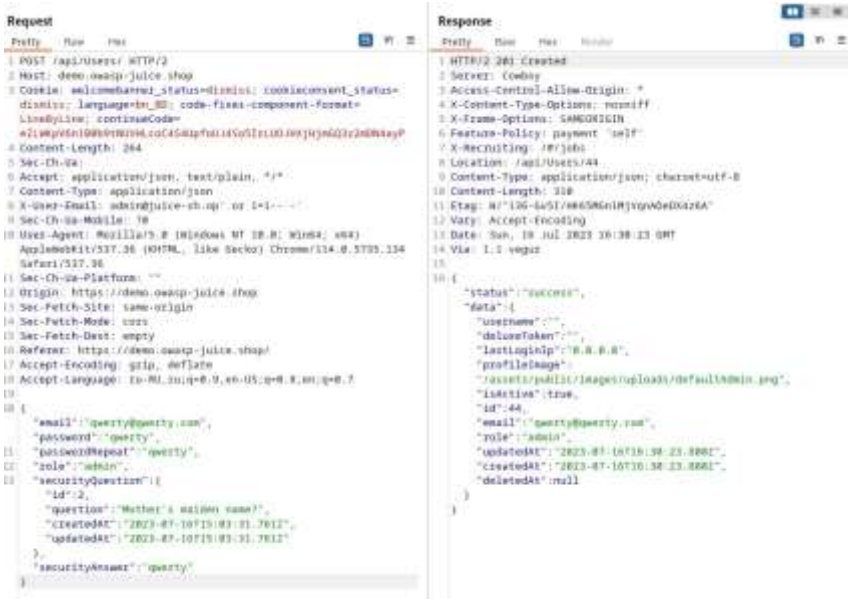
13))Уязвимось назначений роли администратора

Открываем ссылку «https://demo.owasp-juice.shop/#/register»
=> создаем новую учетную запись
=>далее отправляем форому регистрации
=>переходим в инструмент «BurpSuite» на вкалдку «Proxy»
=>находим нужный для нас запрос «POST /api/Users/ HTTP/2»
=>отправляем его в «Repiter»
=> добавлем значение «role: “admin” , »
отправялем запрос.



Screenshot1

Screenshot2



Screenshot3

Remediation Summary

Как результат «» было скомпрометирован вэб-приложение, были найдены критические и очень высокого уроня уязвимости которые необходимо незамедлительно исправить. Большая часть уязвимостей связана с открытыми для сканирования директориями и файлами, непосредственно находящаяся информация на сервере в виде доступных лог файлов, файла бэкапа вэб приложения, с открытой информацией формата «OSINT» и многие другие файлы. Доступность информации это не единственная уязвимость , также присутствует проблема дефолтных паролей, Контроля сессии, уязвимость недостаточный уровень шифрования паролей, возможность загрузки сторонних файлов с изменением формата чтения документа/изображения/скрипта. Также видна проблема в назначении любого пользователя правами администратора и соответствующий уровень доступа

Short Term

[**Insecure Direct Object References**]- Удалить из директорий информацию о бэкапах, и о всех файлах советующие типу «Конфиденциальная информация»

[**Using Default Credentials**]- Используйте сильные пароли и регулярно обновлять их

[**Access Log File Dump**] – Убедитесь, что доступ к файлам журнала доступа (access log files) ограничен, доступ к файлу может быть только у Суперадмина

[**Dictionary Attack**]- Необходимо использовать пароли сложнее и длине и с хешированием пароля более сложной хеш-функцией чем md5, также необходимо использовать дополнительное значение «СОЛЬ» при хешировании паролей и учетных данных.

Medium Term

[**SQL Injection**]- Чтобы предотвратить SQL-инъекции, рекомендуется использовать параметризованные запросы или подготовленные выражения, чтобы обеспечить безопасную обработку пользовательского ввода

[**Using Default Credentials**]- Используйте сильные пароли и регулярно обновляйте их. Рекомендуется также применять механизмы двухфакторной аутентификации (2FA) для повышения безопасности при каждом входе в учетную запись

[**Sensitive Data Exposure**] – Убедитесь, что доступ к файлам журнала доступа (access log files) ограничен только авторизованным пользователям или администраторам

[**NoSQL Injection**] Необходимо проводить правильную вариацию и фильтрацию входных данных, которые передаются в операции базы данных NoSQL

[Manipulation NoSql]-Для защиты от таких атак рекомендуется использовать хорошо спроектированные механизмы аутентификации и авторизации, проверять и фильтровать пользовательский ввод, регулярно обновлять и поддерживать базы данных и следить за обновлениями и рекомендациями от разработчиков баз данных NoSQL.

Long Term

- **Обновление операционных систем и прошивок:**

  Регулярно обновляйте операционные системы и прошивки на серверах и устройствах, чтобы исправить известные уязвимости и обеспечить безопасные настройки по умолчанию.

  Следите за релизами обновлений, патчей и исправлений безопасности от поставщиков операционных систем и производителей устройств.

- **Корректное обращение с конфиденциальными данными:**

  Осуществляйте защиту конфиденциальных данных с использованием шифрования в покое и в движении.

  Правильно управляйте жизненным циклом конфиденциальных данных, включая их сбор, передачу, хранение и уничтожение.

  Ограничьте доступ к конфиденциальным данным только необходимым сотрудникам и учетным записям, и применяйте механизмы контроля доступа.

- **Реализация надежных механизмов аутентификации и управления сеансами:**

  Используйте проверенные методы аутентификации, такие как , OpenID Connect, для обеспечения безопасного входа в систему.

  Управляйте сеансами пользователей, включая правильное установление тайм-аутов, повторную аутентификацию для критических операций и защиту от атак подбора сеансов (session hijacking).

- **Обучение пользователей и персонала:**

  Проводите обучение пользователей по безопасному использованию паролей, аутентификации и обращению с конфиденциальными данными.

  Подготовьте персонал к обнаружению и реагированию на возможные угрозы, связанные с нарушением аутентификации и управлением сеансами.

- **Проведение регулярных аудитов безопасности и тестирования:**

  Периодически проводите аудиты безопасности, чтобы обнаружить слабые места и уязвимости в системе, связанные с нарушением аутентификации и управлением сеансами.

  Проводите регулярное тестирование на проникновение (penetration testing) для выявления уязвимостей и проверки эффективности принятых мер безопасности.

Описание для технического специалиста

1. Sql-injection - P1

CWE	CWE-82
Description	В окно ввда логин было использовано следующее значение «’ or 1=1-- -’» это значение может быть использовано в множествнных местах
Security Impact	Уровень влияния – критический, при помаши этой уязвимости получил доступ к учетной записи администратора, затем получил доступ к всем файлам и директориям имущие соответствующий уровень доступа. Также это уязвимость работает и в полях ввода пороля, сброс пароля, доставка товара.
Affected Domain	https://demo.owasp-juice.shop/#/login
Remediation	Ограничьте права доступа к базе данных, предоставляя только те разрешения, которые необходимы для выполнения операций. Предоставляйте базе данных только минимальный набор привилегий, чтобы снизить потенциальный ущерб при возможной эксплуатации SQL Injection. Регулярное обновление и патчинг: Регулярно обновляйте используемую базу данных и ее драйверы, чтобы исправить известные уязвимости, связанные с SQL Injection. Поставщики баз данных и разработчики постоянно работают над улучшением безопасности и выпускают патчи и обновления для предотвращения эксплуатации уязвимостей.
External References	https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html

2. Using Default Credentials/pass (использование дефолтного пароля для администратора) **- P1**

CWE	CWE-259
Description	Осмотрев Вэб приложение был найден электронный адрес предполагаемого администратора, и использован пароль « admin123»,после чего было произведен вход в учетную запись.
Security Impact	Эта уязвимость возникает, когда аутентификационные данные (логин и пароль) для доступа к системе или ресурсу используются по умолчанию или остаются неизменными после установки. Злоумышленники могут легко получить доступ к системе или ресурсу с использованием этих стандартных учетных данных.
Affected Domain	https://demo.owasp-juice.shop/#/login
Remediation	Измените все стандартные учетные данные по умолчанию после установки системы или ресурса. - Используйте сильные и уникальные пароли для аутентификации. - Регулярно изменяйте пароли и избегайте повторного использования паролей. - Используйте механизмы двухфакторной аутентификации (2FA) для дополнительного слоя защиты.
External References	https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/04-Authentication_Testing/02-Testing_for_Default_Credentials

3. Manipulation NoSql injection/Command Injection. (получили всю базу данных пользователей и Администраторов) **- P1**

CWE	**CWE-77**
Description	Переходим по ссылке «https://demo.owasp-juice.shop» выполнеям команду поиска «apple» предварительно перехватив запрос через через «Proxy>Intercept» отправляем запрос в «Repiter», меняем значение : «/rest/products/search?q=apple» на следующее значение «/rest/products/search?q=rest/products/searh?q=%27))%20union%20select%20id,email,password,4,5,6,7,8,9%20from%20users--» Получаем логины и пароли всех пользователей.
Security Impact	Эта уязвимость возникает, когда злоумышленник может внедрить вредоносный код или команды в запросы, предназначенные для баз данных или операционной системы. В результате злоумышленник может получить несанкционированный доступ к данным, выполнить произвольные команды или контролировать систему.
Affected Domain	https://demo.owasp-juice.shop/#/serach?q=
Remediation	Не доверяйте пользовательскому вводу и обязательно проводите проверку и фильтрацию данных перед использованием их в запросах. - Используйте параметризованные запросы или подготовленные выражения для обработки пользовательского ввода. - Примените контекстуальное экранирование или фильтрацию данных, чтобы предотвратить интерпретацию пользовательского ввода как команды или кода. - Ограничьте привилегии доступа к базе данных и операционной системе, предоставляя только необходимые права.
External References	https://owasp.org/www-pdf-archive/GOD16-NOSQL.pdf

4. Users credentials (Получили доступ к логам всех пользователей за период) **- P1**

CWE	**CWE-532**
Description	Запускаем сканет «FFUF» получаем перечен отсканированых дерикторий и файлов. Переходим по ссылке «https://demo.owasp-juice.shop/support/log» и видим файл открываем его через дополнительную команду «https://demo.owasp-juice.shop/support/log/access.log.2023-07-14%25%30%30.md» Получен файл с логами всех пользователей
Security Impact	Эта уязвимость возникает, когда логи системы или приложения содержат хранящиеся в них пользовательские учетные данные, такие как пароли или личная информация. Если злоумышленник получает несанкционированный доступ к этим логам, это может привести к компрометации пользовательских учетных данных и возможному злоупотреблению этой информацией.
Affected Domain	https://demo.owasp-juice.shop/support/logs
Remediation	Никогда не храните пользовательские учетные данные в логах системы или приложения. Если логи уже содержат такую информацию, удалите ее или примените механизмы шифрования или хеширования для обеспечения безопасного хранения. - Установите строгие контроли и ограничения доступа к логам системы, чтобы предотвратить несанкционированный доступ. - Мониторьте логи системы и приложения на наличие аномалий и подозрительной активности.
External References	https://owasp.org/www-project-top-ten/2017/A2_2017-Broken_Authentication

5. Users Credentials Log Access Dump - P1

CWE	CWE-532
Description	Переходим по ссылке «https://demo.owasp-juice.shop/ftp/support/log» и видим файл с конфиденциальной информацией.
Security Impact	Эта уязвимость возникает, когда логи системы или приложения содержат хранящиеся в них пользовательские учетные данные, такие как пароли или личная информация. Если злоумышленник получает несанкционированный доступ к этим логам, это может привести к компрометации пользовательских учетных данных и возможному злоупотреблению этой информацией.
Affected Domain	https://demo.owasp-juice.shop/ftp/support/log
Remediation	Никогда не храните пользовательские учетные данные в логах системы или приложения. Если логи уже содержат такую информацию, удалите ее или примените механизмы шифрования или хеширования для обеспечения безопасного хранения. - Установите строгие контроли и ограничения доступа к логам системы, чтобы предотвратить несанкционированный доступ. - Мониторьте логи системы и приложения на наличие аномалий и подозрительной активности.
External References	https://owasp.org/Top10/A01_2021-Broken_Access_Control/

6. Gained Access to a File with Server Backups

CWE	CWE-533
Description	Переходим по ссылке «https://demo.owasp-juice.shop/ » выполняем поиск «apple » и отставляем отзыв о товаре «DmitryPavlov_good_teacher» открываем инструмент «BurpSute» выполняем процесс перехвата запроса через «Proxy>Intercept» отправляем запрос в «Repiter», в самом запросе замещаем «"message":"dsadasdsdadas",«"author":"admin@juice-sh.op"» на «{"message":"DmitryPavlov_good_teacher","id":{"$ne":-1}}» и замещаем «POST /rest/products/1/reviews» на «PATCH /rest/products/reviews» отправляем запрос на сервер обратно.
Security Impact	Эта уязвимость возникает, когда злоумышленник получает несанкционированный доступ к файлу с резервными копиями сервера. Если резервные копии содержат конфиденциальную информацию или критические данные, это может привести к утечке информации, нарушению конфиденциальности и возможной потере данных.
Affected Domain	https://demo.owasp-juice.shop/search?q*
Remediation	Убедитесь, что файлы с резервными копиями сервера хранятся в безопасном и недоступном для несанкционированного доступа месте. Регулярно проверяйте и обновляйте права доступа к файлам с резервными копиями, чтобы предотвратить несанкционированный доступ. Используйте механизмы шифрования или другие меры безопасности для защиты файлов с резервными копиями от несанкционированного чтения или использования. Применяйте политику управления резервными копиями, включая их периодическую проверку, удаление устаревших копий и обеспечение их безопасного хранения.
External References	https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/02-Configuration_and_Deployment_Management_Testing/04-Review_Old_Backup_and_Unreferenced_Files_for_Sensitive_Information

7. Unauthorized Privilege Assignment (была создана учетноя запись и наделена правми администратора) **- P1**

CWE	**CWE-269**
Description	Открываем ссылку «https://demo.owasp-juice.shop/#/register» => создаем новую учетную запись =>далее отправляем форому регистрации =>переходим в инструмент «BurpSuite» на вкалдку «Proxy» =>находим нужный для нас запрос «POST /api/Users/ HTTP/2» =>отправляем его в «Repiter» => добавлем значение «role: “admin” , » отправялем запрос.
Security Impact	Уязвимость возникает, когда административные привилегии или роль администратора назначаются на учетную запись без соответствующих прав. Злоумышленник может получить несанкционированный доступ к привилегированным функциям или ресурсам, что может привести к возможности выполнения недопустимых действий или компрометации системы.
Affected Domain	https://demo.owasp-juice.shop/#/register/
Remediation	Разработайте и примените строгую политику назначения привилегий, чтобы убедиться, что только соответствующим пользователям или ролям предоставляются привилегии администратора. Ограничьте доступ к административным функциям и ресурсам только авторизованным пользователям с необходимыми привилегиями. Установите контроль доступа на уровне ролей и регулярно проверяйте и обновляйте привилегии, чтобы избежать несанкционированного повышения привилегий.
External References	https://owasp.org/API-Security/editions/2023/en/0xa3-broken-object-property-level-authorization/