DSEC by SOLAR

DSEC.RU

Отчет

о результатах повторной проверки наличия уязвимостей компонента «Общественное голосование» Платформы обратной связи (ПОС)

Даты проведения работ:

23.12.2025

Технический менеджер проекта:

А.С. Сидуков

Директор департамента аудита:

В. Н. Акименко

КОНФИДЕНЦИАЛЬНО

Содержание

1. ВВЕДЕНИЕ 3

1.1. Информация о проекте 3
1.2. Цель работ 3
1.3. Детали проверки 3
1.4. Область работ 3

2. КРАТКОЕ ОПИСАНИЕ ПРОВЕДЕННЫХ РАБОТ 4

2.1. Результаты работ 4

3. ПЕРЕЧЕНЬ УЯЗВИМОСТЕЙ 6

3.1. Недостатки контроля доступа 6
3.2. Небезопасные прямые ссылки на объекты, раскрывающие персональные данные пользователей 14
3.3. Небезопасные прямые ссылки на объекты, предоставляющие доступ к загруженным файлам и информации о проектах 17
3.4. Межсайтовый скриптинг отраженный (Reflected XSS) 23
3.5. Возможность проведения атаки «Подделка запроса на стороне сервера (SSRF)» 26
3.6. Небезопасная конфигурация механизма кэширования 28
3.7. Недостатки бизнес-логики приложения 31
3.8. Небезопасные прямые ссылки на объекты, позволяющие получить данные уведомлений и документов 35
3.9. Возможность проведения атаки «Внедрение HTML-кода на страницу веб-приложения» 38

4. ПЕРЕЧЕНЬ НЕДОСТАТКОВ 41

4.1. Раскрытие отладочной и конфигурационной информации 41
4.2. Некорректная обработка ошибок 43
4.3. Некорректные HTTP-заголовки безопасности 45
4.4. Использование тестовых данных в продуктовой среде 47
4.5. Раскрытие чувствительной информации 51

ПРИЛОЖЕНИЕ 1. МЕТОДИКА ОЦЕНКИ ЗАЩИЩЕННОСТИ 54

1.Введение

1.1. Информация о проекте

Настоящий отчет содержит результаты работ по повторной проверке наличия уязвимостей компонента «Общественное голосование» Платформы обратной связи (ПОС).

Работы проводились в соответствии с ГК №071/25/61 от 30.05.2025 специалистами компании ООО «РТК ИБ» (далее – Исполнитель), дата проведения работ: 23.12.2025.

1.2. Цель работ

Целью работ является повторная проверка наличия уязвимостей и недостатков приложения после выполнения Заказчиком рекомендаций по устранению.

1.3. Детали проверки

В качестве модели нарушителя рассматривались следующие модели:

1. **[H1]**: Нарушитель, имеющий сетевой доступ к приложению, не обладающий какой-либо информацией о нем. Может самостоятельно регистрироваться (при наличии механизмов саморегистрации) и использовать стандартную функциональность.
2. **[H2]**: Нарушитель, имеющий сетевой доступ к приложению, обладающий информацией о нем (например, описание программных интерфейсов приложения, схема архитектуры) и учетной записью пользователя с определенной ролью в приложении (в том числе и административной).

Методика оценки защищенности описана в Приложении 1 к настоящему отчету.

1.4. Область работ

Область проведения работ по повторной проверке ограничена веб-ресурсами **pos.gosuslugi.ru/og/**, **pos.gosuslugi.ru/lkp/** и **fkgs.gosuslugi.ru**.

2.Краткое описание проведенных работ

2.1. Результаты работ

Проведенная повторная проверка наличия уязвимостей и недостатков показала, что обнаруженные ранее уязвимости, позволяющие проводить атаки «Подделка запроса на стороне сервера (SSRF)» и «Внедрение HTML-кода на страницу веб-приложения», были успешно устранены. Также были исправлены недостатки бизнес-логики приложения и недостатки, связанные с некорректной обработкой ошибок.

Кроме того, были частично исправлены уязвимости «Небезопасные прямые ссылки на объекты, предоставляющие доступ к загруженным файлам и информации о проектах». Так, для указанной уязвимости были исправлены все сценарии, кроме сценария **/og/document/file?file_info_id=773999**, позволяющего авторизованному пользователю получить загруженные в приложение файлы.

Также были частично исправлены недостатки контроля доступа, а именно были исправлены сценарии **/og/backend/api/v1/poll/send-to-gp** и **/og/backend/api/v1/users/roles**. Однако при этом остальные сценарии исправлены не были. Так, недостатки контроля доступа по-прежнему позволяют авторизованному пользователю без административных прав получить список пользователей ЕСИА, включая их личные email-адреса, а также список вложений пользователей. Также по-прежнему не исправлены сценарии, позволяющие редактировать и удалять вложения других пользователей.

Помимо прочего, не были устранены уязвимости «Небезопасные прямые ссылки на объекты, раскрывающие персональные данные пользователей». Так, неавторизованный пользователь по-прежнему может просматривать файлы из хранилища, такие как загруженные документы или скриншоты пользователей, в которых могут содержаться персональные данные.

Также не были исправлены уязвимости, позволяющие проводить атаки «Межсайтовое внедрение сценариев (XSS)», уязвимости, связанные с небезопасной конфигурацией механизма кэширования, и уязвимости «Небезопасные прямые ссылки на объекты, позволяющие получить данные уведомлений и документов».

Проведенная повторная проверка также показала, что не были устранены недостатки, приводящие к раскрытию отладочной и конфигурационной информации, а также к раскрытию чувствительной информации. Кроме того, не были устранены недостатки, связанные с использованием тестовых данных в продуктовой среде и некорректными HTTP-заголовками безопасности.

Подробная информация о результатах проверки приведена в таблицах 1 и 2.

Таблица 1. Результаты проверки исправления уязвимостей

Пункт	Уязвимость	Риск	Статус
3.1	Недостатки контроля доступа	**Высокий**	**Частично исправлено**
3.2	Небезопасные прямые ссылки на объекты, раскрывающие персональные данные пользователей	**Высокий**	**Не исправлено**

Пункт	Уязвимость	Риск	Статус
3.3	Небезопасные прямые ссылки на объекты, предоставляющие доступ к загруженным файлам и информации о проектах	Средний	Частично исправлено
3.4	Межсайтовый скриптинг отраженный (Reflected XSS)	Средний	Не исправлено
3.5	Возможность проведения атаки «Подделка запроса на стороне сервера (SSRF)»	Средний	Исправлено
3.6	Небезопасная конфигурация механизма кэширования	Низкий	Не исправлено
3.7	Недостатки бизнес-логики приложения	Низкий	Исправлено
3.8	Небезопасные прямые ссылки на объекты, позволяющие получить данные уведомлений и документов	Низкий	Не исправлено
3.9	Возможность проведения атаки «Внедрение HTML-кода на страницу веб-приложения»	Низкий	Исправлено

Таблица 2. Результаты проверки исправления недостатков

Пункт	Недостаток	Статус
4.1	Раскрытие отладочной и конфигурационной информации	Не исправлено
4.2	Некорректная обработка ошибок	Исправлено
4.3	Некорректные HTTP-заголовки безопасности	Не исправлено
4.4	Использование тестовых данных в продуктовой среде	Не исправлено
4.5	Раскрытие чувствительной информации	Не исправлено

3.Перечень уязвимостей

3.1. Недостатки контроля доступа

Статус: ***Частично исправлено***
Критичность: ***Высокая***
Вероятность эксплуатации: ***Высокая***
Итоговый риск: ***Высокий***
Модель нарушителя: [H2]

Результат перепроверки:

Сценарии **/og/backend/api/v1/poll/send-to-gp** и **/og/backend/api/v1/users/roles** не воспроизводились. Остальные сценарии исправлены не были.

Описание:

контроль доступа устанавливает ограничения на выполнение определенных действий и получение доступа к ресурсам для пользователей с различным уровнем привилегий. Механизм контроля доступа предназначен для защиты данных от несанкционированного доступа и ограничения использования привилегированной функциональности. Однако в исследованном приложении отсутствует или некорректно реализована проверка прав доступа, что позволяет пользователю совершать действия вне установленных для него привилегий. Таким образом, потенциальный злоумышленник может использовать недостатки контроля доступа для компрометации чувствительных данных или получения дополнительных функциональных возможностей.

Риск:

- получение доступа к данным других пользователей;
- чтение и изменение обрабатываемой в приложении информации;
- выполнение действий, недоступных пользователю из графического интерфейса.

Технические детали:

Уязвимые хосты:

- pos.gosuslugi.ru

В приложении было обнаружено большое количество недостатков контроля доступа, подробная информация о которых приведена в таблице ниже (см. Таблица 3).

Таблица 3. Сведения об обнаруженных недостатках контроля доступа

Сценарии	Описание
/og/document/	Авторизованный пользователь без административных прав может получить список документов веб-приложения.

Сценарии	Описание
/og/person/	Авторизованный пользователь без административных прав может получить список пользователей ЕСИА, включая их личные email-адреса, а также проверить статус подписки.
/og/attachment/	Авторизованный пользователь без административных прав может получить список вложений пользователей веб-приложения.
/og/attachment/update /og/attachment/delete	Возможность редактирования и удаления вложений других пользователей.
/og/backend/api/v1/poll/send-to-gp	В приложении обнаружена возможность отправки данных опроса других пользователей, доступ к которым отсутствует, в госпаблики. Функциональность доступна для опросов в статусе «В процессе». Параметр entityId – идентификатор опроса.
/og/backend/api/v1/users/roles	Авторизованному пользователю доступен список ролей пользователей веб-приложения.

Примеры эксплуатации:

Пример получения авторизованным пользователем без административных прав списка документов веб-приложения:

```
GET /og/document HTTP/1.1
Host: pos.gosuslugi.ru
Cookie: _identity-backend=t9p46obp75s3npa6lvfr1orbr5
```

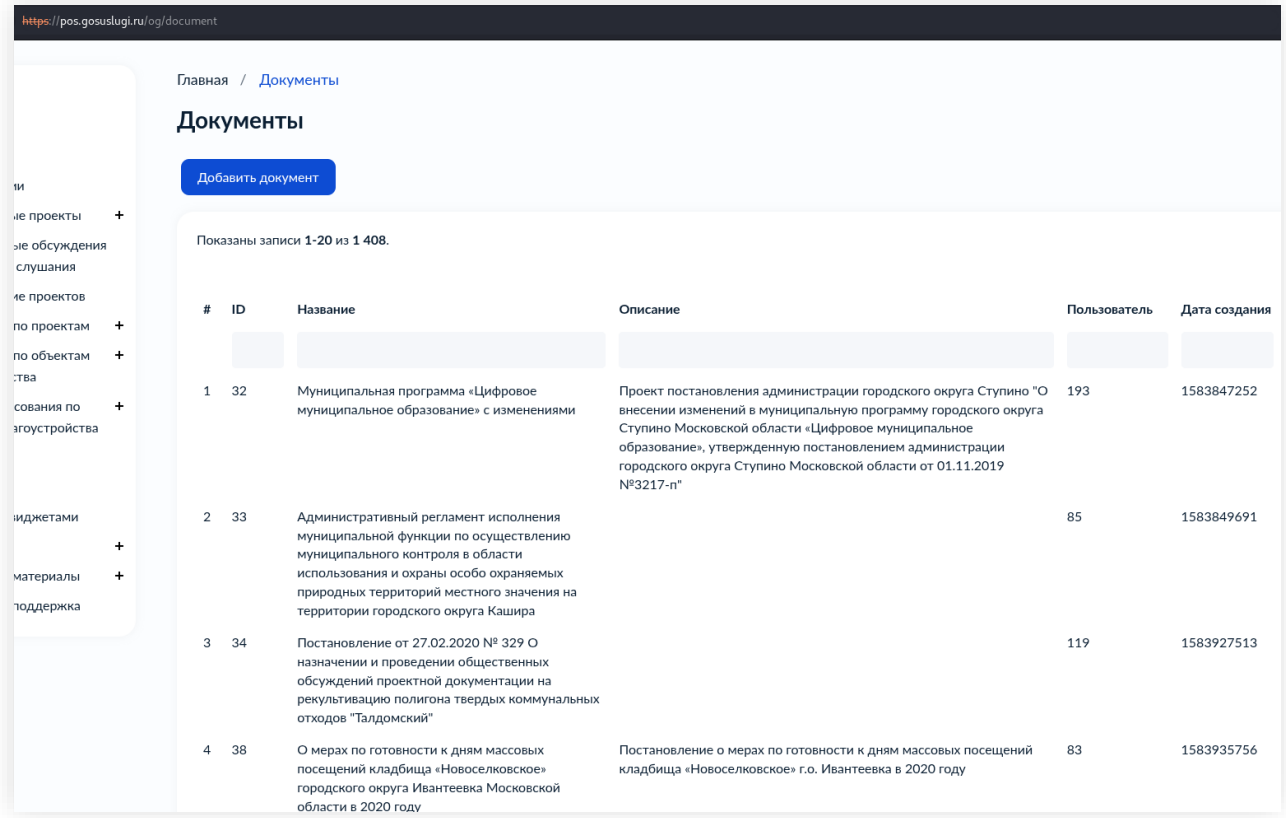

Рисунок 1. Доступ к списку документов веб-приложения

Пример получения авторизованным пользователем без административных прав списка пользователей ЕСИА, включая их личные email-адреса и статус подписки:

```
GET /og/person HTTP/1.1
Host: pos.gosuslugi.ru
Cookie: _identity-backend=t9p46obp75s3npa6lvfr1orbr5
```

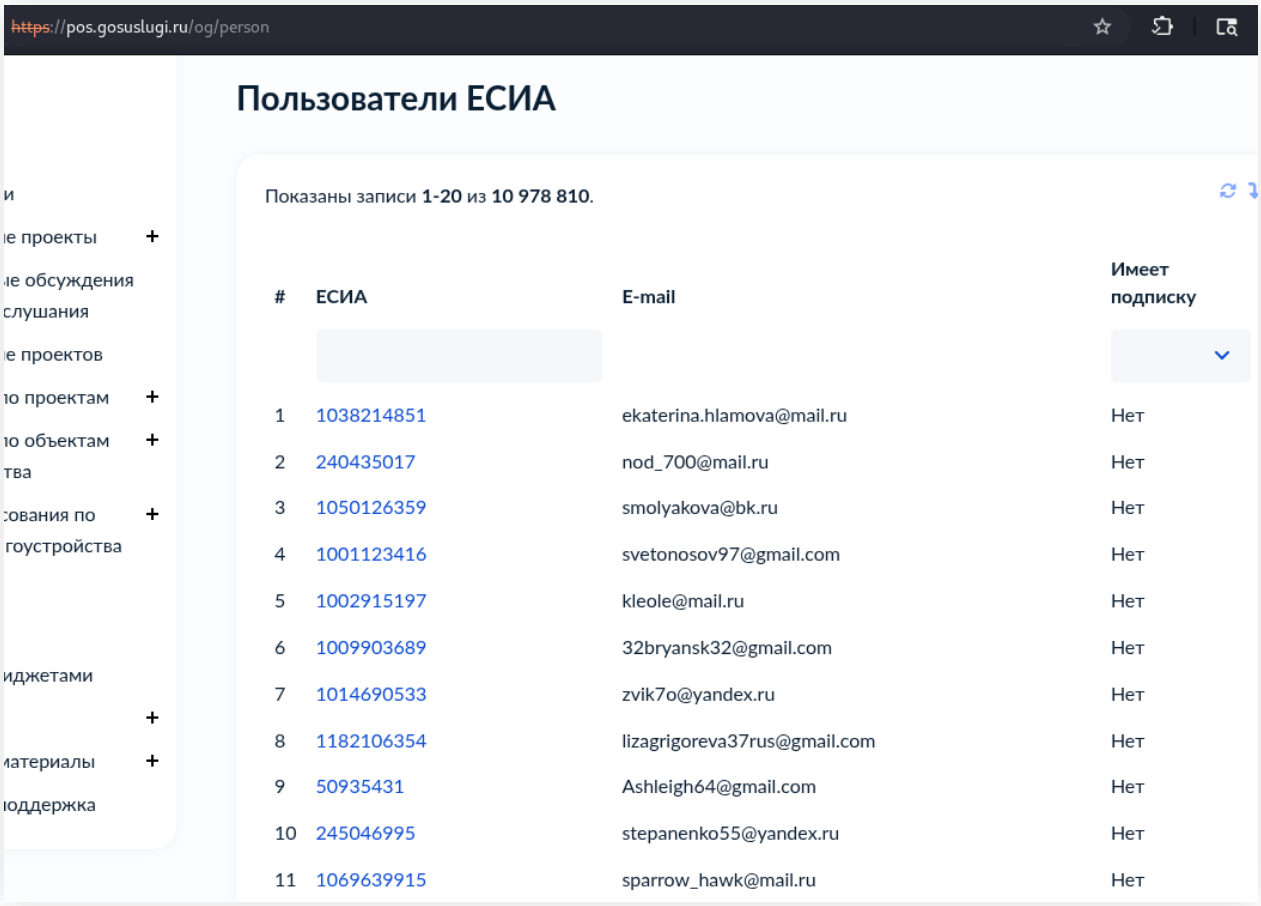

Рисунок 2. Доступ к списку пользователей ЕСИА

Пример получения авторизованным пользователем без административных прав списка вложений пользователей веб-приложения:

```
GET /og/attachment HTTP/1.1
Host: pos.gosuslugi.ru
Cookie: _identity-backend=t9p46obp75s3npa6lvfr1orbr5
```

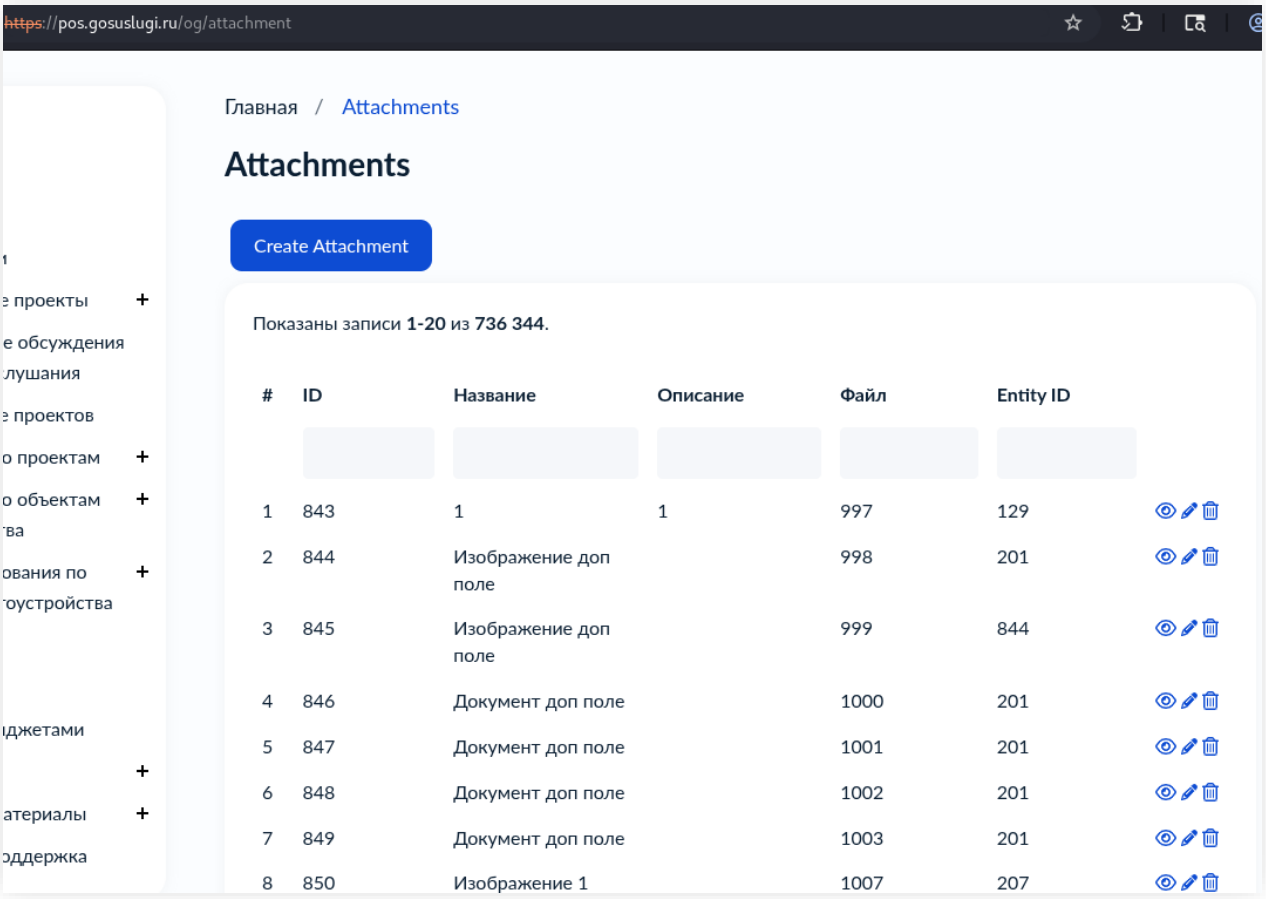

Рисунок 3. Доступ к списку вложений веб-приложения

Пример редактирования вложений других пользователей:

```
POST /og/attachment/update?id=1112 HTTP/1.1
Host: pos.gosuslugi.ru
Cookie: <COOKIE>
Content-Length: 287
Content-Type: application/x-www-form-urlencoded

_csrf-
backend=<TOKEN>&Attachment%5Btitle%5D=testt&Attachment%5Bdescription%5D=&Attach
ment%5Bfile_info_id%5D=1559&Attachment%5Bentity_id%5D=96&Attachment%5Bentity_ty
pe%5D=common%5Cmodels%5CProjectContest
```

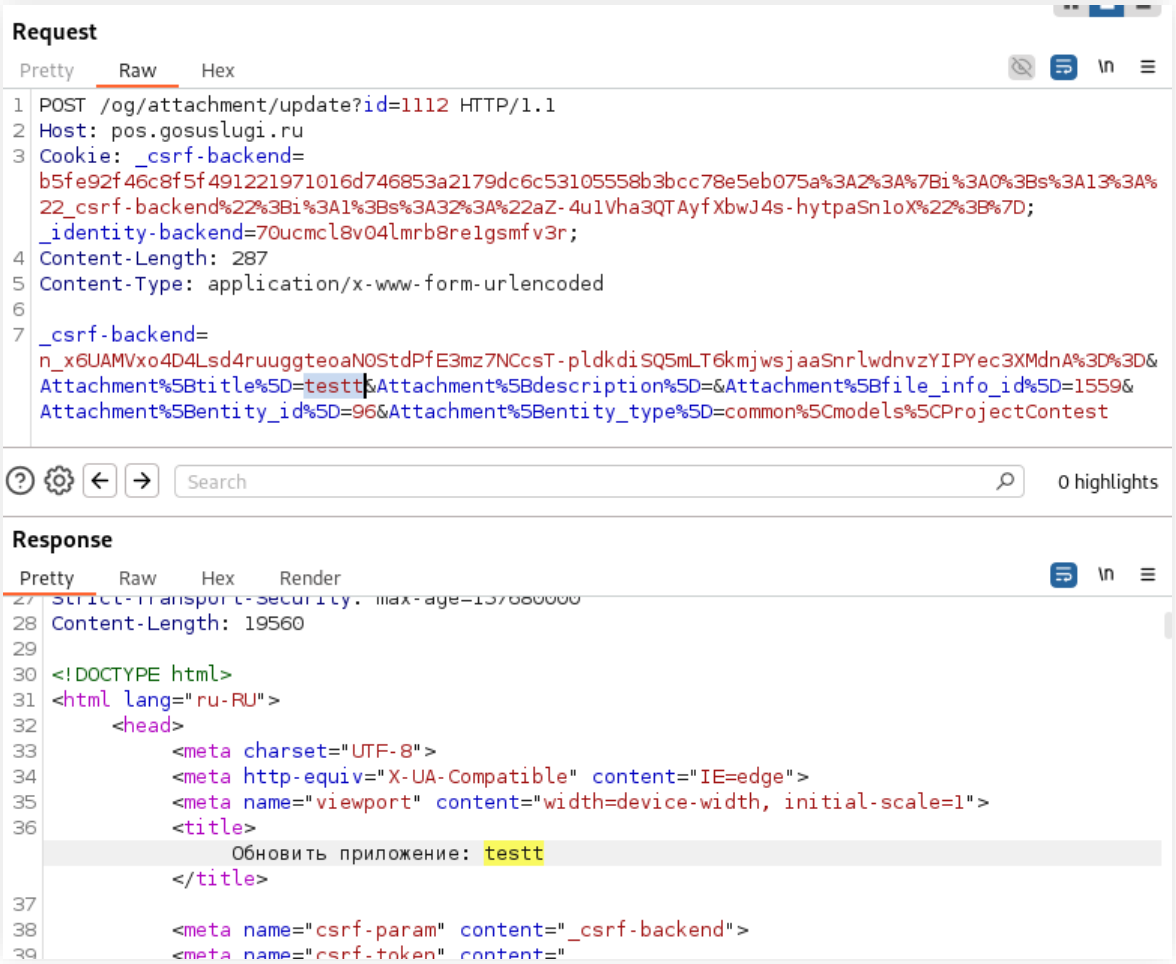

Рисунок 4. Редактирование вложения другого пользователя

Пример удаления вложений других пользователей:

```
DELETE /og/attachment/delete?id=168396 HTTP/1.1
Host: pos.gosuslugi.ru
Cookie: <COOKIE>
```

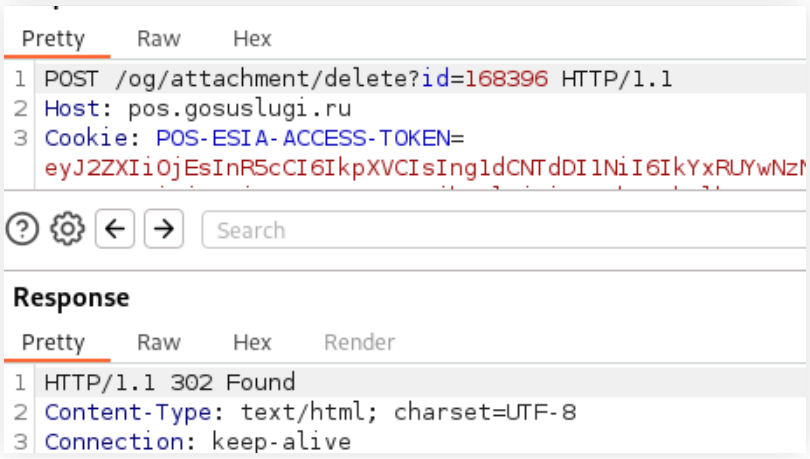

Рисунок 5. Удаление вложения другого пользователя

Пример отправки данных опроса другого пользователя в госпаблики:

```
POST /og/backend/api/v1/poll/send-to-gp HTTP/1.1
Host: pos.gosuslugi.ru
Cookie: <COOKIE>
Content-Length: 39
Content-Type: application/json
Connection: keep-alive

{"entityId":490003,"entityType":"poll"}
```

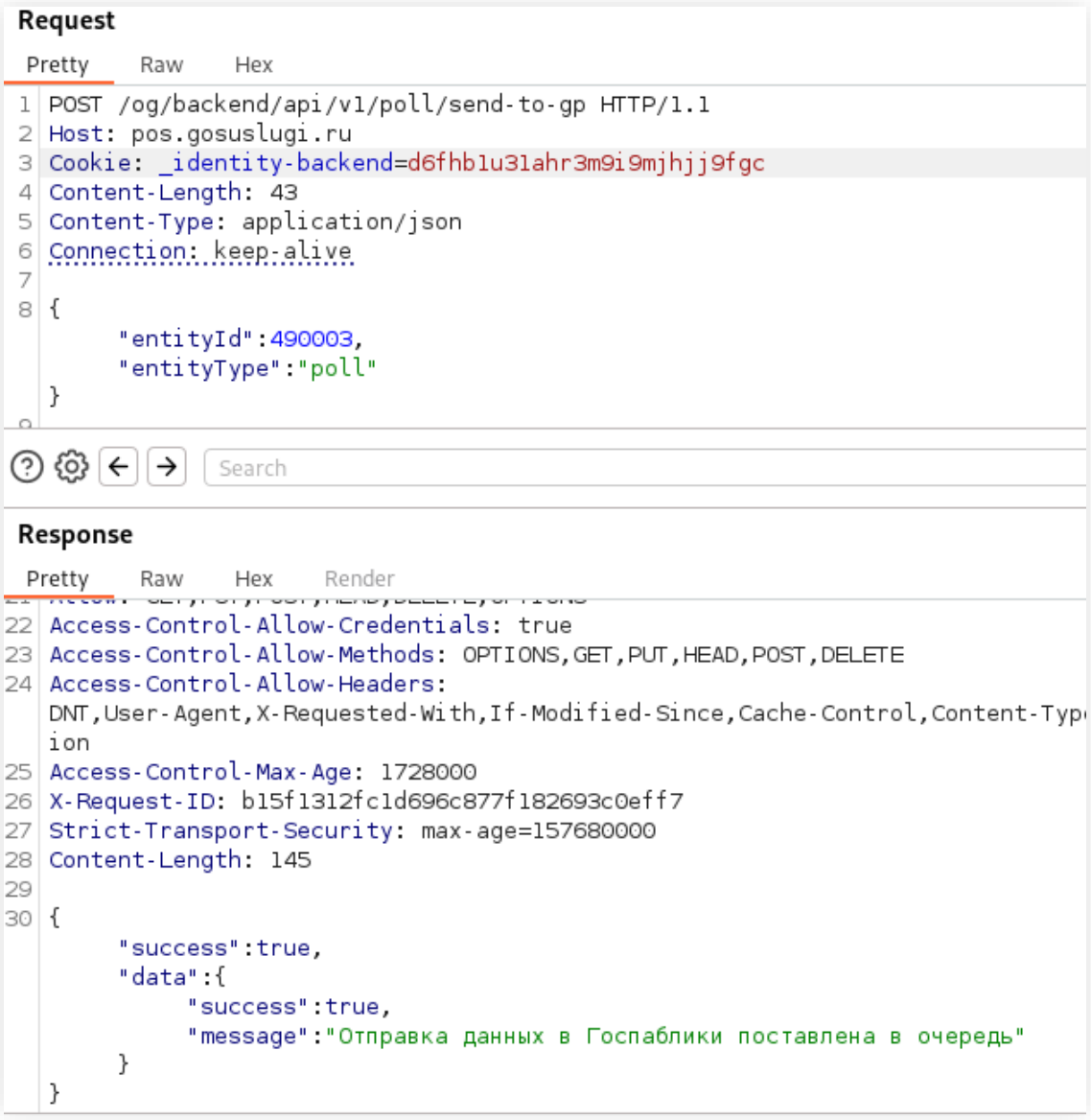

Рисунок 6. Пример отправки данных опроса другого пользователя в госпаблики

Пример получения авторизованным пользователем списка ролей пользователей веб-приложения:

```
GET /og/backend/api/v1/users/roles HTTP/1.1
Host: pos.gosuslugi.ru
Cookie: <COOKIE>
```

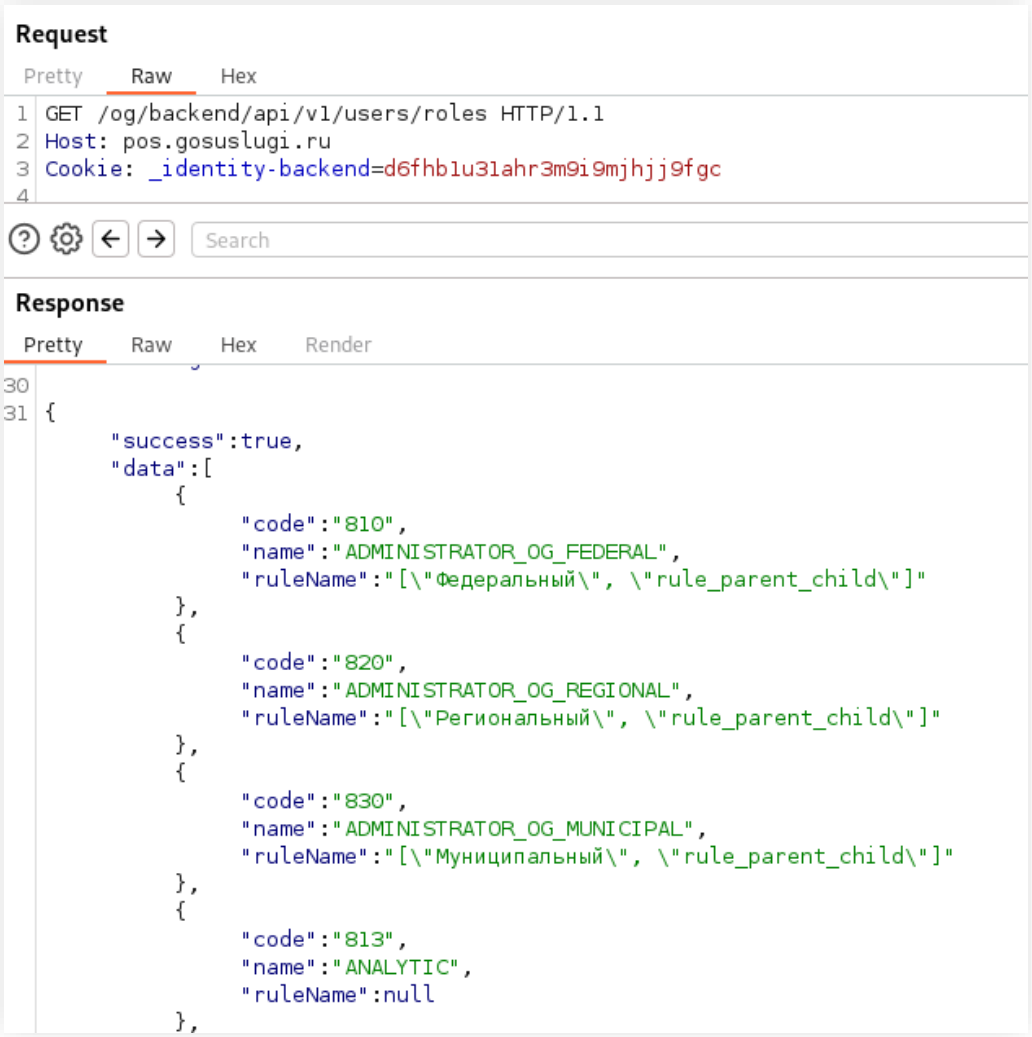

Рисунок 7. Доступ к списку ролей пользователей веб-приложения

Рекомендации:

- Реализовать эффективное разграничение доступа к ресурсам:
  - ограничить доступ к информации или функциональности при прямом обращении к страницам и объектам приложения;
  - исключить кэширование страниц, содержащих конфиденциальную информацию;
  - разрешать доступ к защищаемым данным только после прохождения процедуры аутентификации.
- Проверять на стороне сервера привилегии пользователя до предоставления ему доступа к данным и функциональности приложения.
- Рассмотреть возможность внедрения библиотек и фреймворков, направленных на управление аутентификацией и авторизацией пользователей.
- Более подробные рекомендации по безопасной настройке контроля доступа можно найти по адресу: https://cheatsheetseries.owasp.org/cheatsheets/Authorization_Cheat_Sheet.html.

3.2. Небезопасные прямые ссылки на объекты, раскрывающие персональные данные пользователей

Статус: ***Не исправлено***
Критичность: ***Высокая***
Вероятность эксплуатации: ***Высокая***
Итоговый риск: ***Высокий***
Модель нарушителя: [H1]

Результат перепроверки:

Примеры воспроизводятся без изменений.

Описание:

небезопасные прямые ссылки на объекты являются одним из типов недостатков контроля доступа. Взаимодействие с данными в приложении реализовано на основе контролируемого пользователем идентификатора. При этом в приложении отсутствует или некорректно реализована проверка наличия у пользователя прав доступа к запрашиваемым объектам. Таким образом, потенциальный злоумышленник может подобрать идентификаторы или использовать идентификаторы элементов, принадлежащих другим пользователям, для получения несанкционированного доступа к данным. Стоит отметить, что инкрементальные идентификаторы упрощают проведение атаки, поскольку являются предсказуемыми и легко подбираемыми.

Риск:

- получение доступа к данным других пользователей;
- чтение обрабатываемой в приложении информации;
- выполнение действий, недоступных пользователю из графического интерфейса.

Технические детали:

Уязвимые хосты:

- pos.gosuslugi.ru

Неавторизованный пользователь может просматривать файлы из хранилища, такие как загруженные документы или скриншоты пользователей. Стоит отметить, что в таких файлах могут содержаться персональные данные пользователей.

Сценарии:

- `/og/storage/pos/attachment/<ID>`
- `/og/storage/pos/images/document/<ID>`

Примеры эксплуатации:

Пример просмотра документа, загруженного другим пользователем:

```
https://pos.gosuslugi.ru/og/storage/pos/attachment/66bdd89075dd8.pdf
```

Пример просмотра скриншота, загруженного другим пользователем:

> `https://pos.gosuslugi.ru/og/storage/pos/images/document/654a156f4f25a`

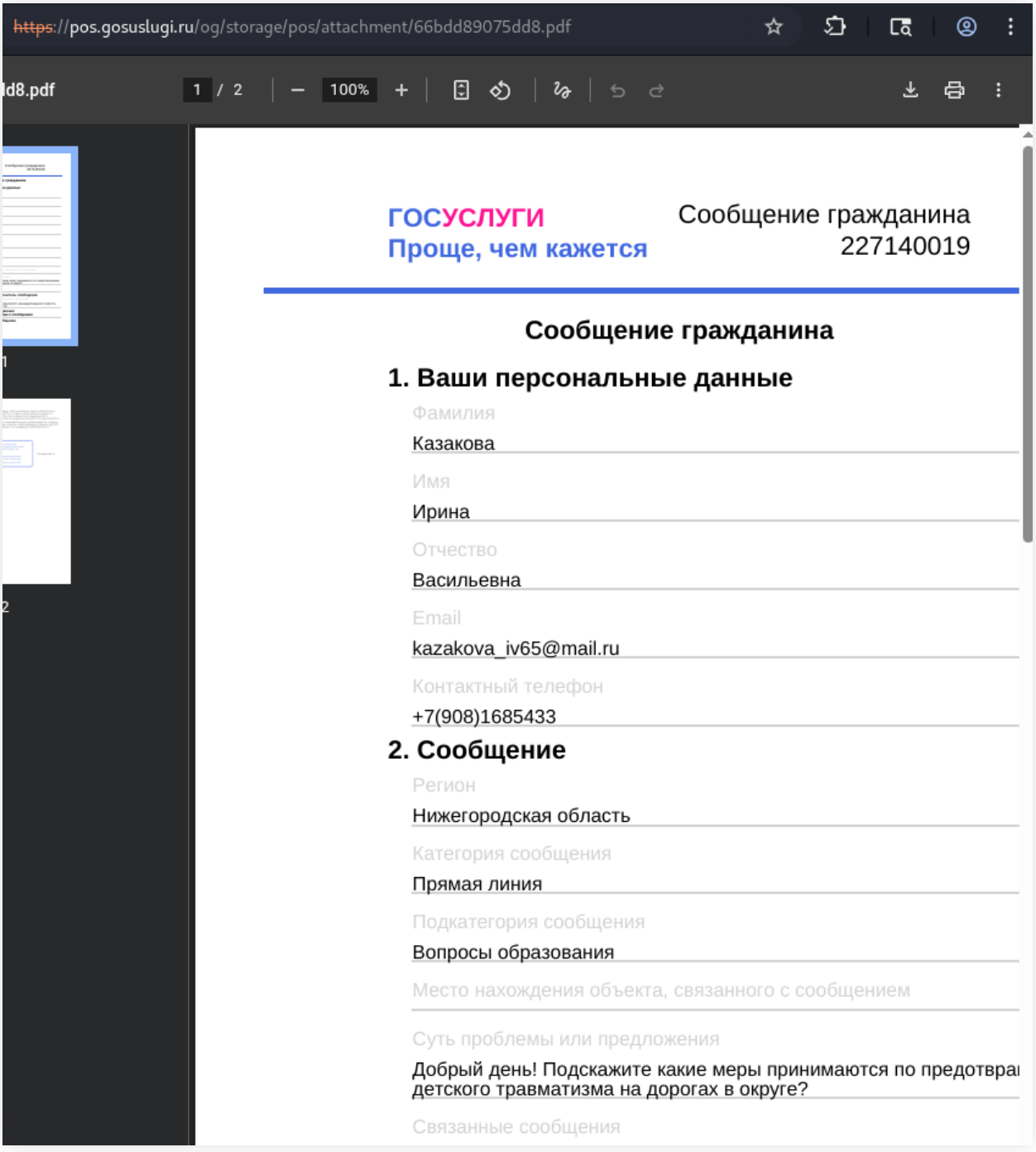

Рисунок 8. Пример просмотра документа, загруженного другим пользователем

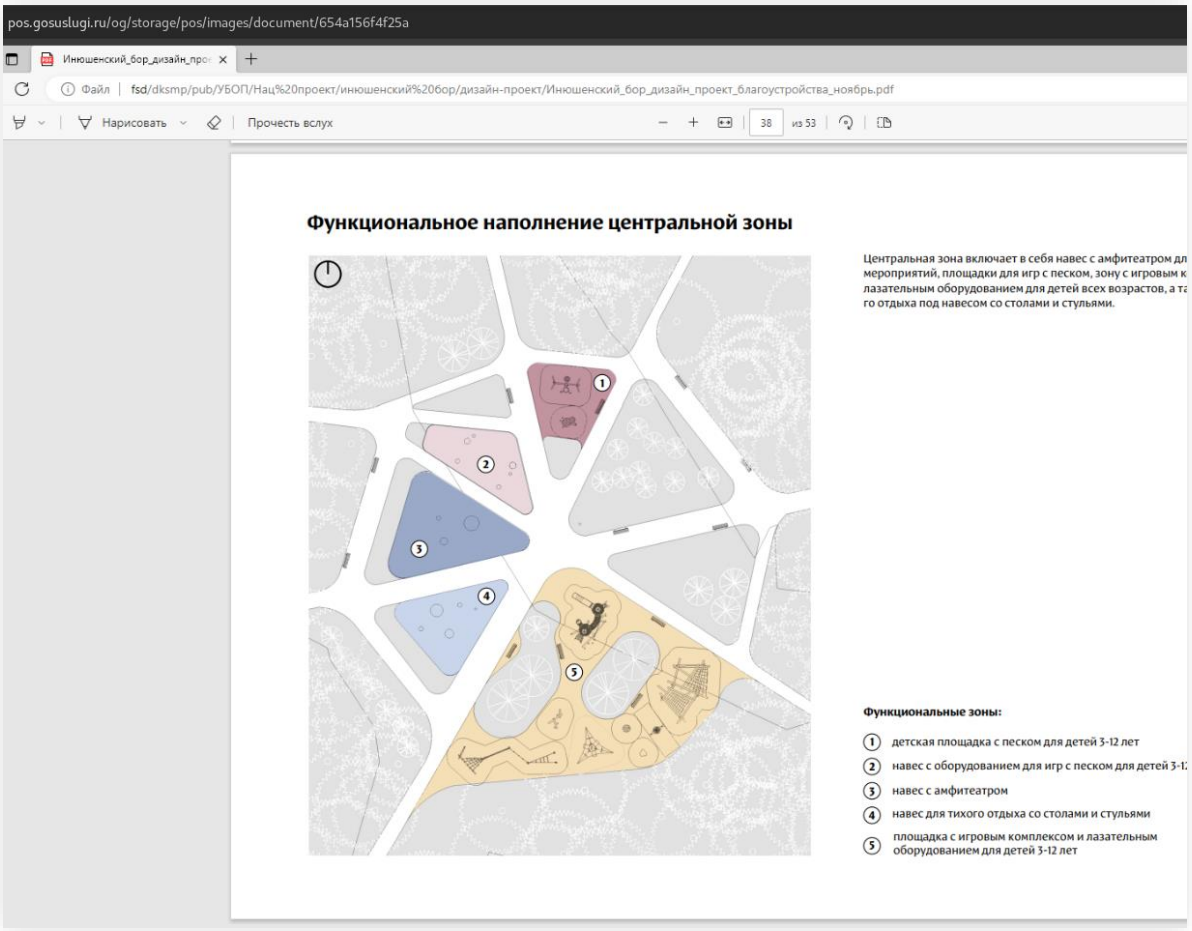

Рисунок 9. Пример просмотра скриншота, загруженного другим пользователем

Рекомендации:

- Реализовать эффективное разграничение доступа к ресурсам:
  - ограничить доступ к информации или функциональности при прямом обращении к страницам и объектам приложения;
  - исключить кэширование страниц, содержащих конфиденциальную информацию;
  - разрешать доступ к защищенным ресурсам только после прохождения процедуры аутентификации.
- Проверять на стороне сервера привилегии пользователя до предоставления ему доступа к данным и функциональности приложения.
- Использовать устойчивые к атаке полного перебора идентификаторы объектов, например GUID.
- Рассмотреть возможность внедрения и использования библиотек и фреймворков, направленных на управление аутентификацией и авторизацией пользователей. Например, для языка Java – JAAS Authorization Framework (https://docs.oracle.com/javase/7/docs/technotes/guides/security/jaas/JAASRefGuide.html) или OWASP Enterprise Security API (ESAPI) (https://owasp.org/www-project-enterprise-security-api/).

- Более подробные рекомендации по безопасной настройке контроля доступа можно найти по адресу: https://cheatsheetseries.owasp.org/cheatsheets/Authorization_Cheat_Sheet.html.

3.3. Небезопасные прямые ссылки на объекты, предоставляющие доступ к загруженным файлам и информации о проектах

Статус: ***Частично исправлено***
Критичность: ***Средняя***
Вероятность эксплуатации: ***Средняя***
Итоговый риск: ***Средний***
Модель нарушителя: [H2]

Результат перепроверки:

Не исправлен сценарий **/og/document/file?file_info_id=773999**. Остальные сценарии не воспроизводятся.

Описание:

небезопасные прямые ссылки на объекты являются одним из типов недостатков контроля доступа. Взаимодействие с данными в приложении реализовано на основе контролируемого пользователем идентификатора. При этом в приложении отсутствует или некорректно реализована проверка наличия у пользователя прав доступа к запрашиваемым объектам. Таким образом, потенциальный злоумышленник может подобрать идентификаторы или использовать идентификаторы элементов, принадлежащих другим пользователям, для получения несанкционированного доступа к данным. Стоит отметить, что инкрементальные идентификаторы упрощают проведение атаки, поскольку являются предсказуемыми и легко подбираемыми.

Риск:

- получение доступа к данным других пользователей;
- чтение обрабатываемой в приложении информации;
- выполнение действий, недоступных пользователю из графического интерфейса.

Технические детали:

Уязвимые хосты:

- pos.gosuslugi.ru

В ходе работ был обнаружен ряд уязвимостей «Небезопасные прямые ссылки на объекты», подробная информация о которых приведена в таблице ниже (см. Таблица 4).

Таблица 4. Сведения об обнаруженных небезопасных прямых ссылках на объекты

Сценарии	Параметры	Описание
/og/project/view	id	Авторизованному пользователю, которому не доступен список проектов, доступен просмотр информации о проекте по его id

Сценарии	Параметры	Описание
/og/project/view-winner	id	Авторизованному пользователю, которому не доступен список проектов из раздела «Проекты-победители», доступен просмотр информации о проекте по его id
/og/initiative-budgeting/view	id	Авторизованному пользователю, которому не доступен список инициативных проектов, доступен просмотр информации о проекте по его id
/og/file-info/download /og/document/file	file_info_id	Авторизованный пользователь может получить загруженные в приложение файлы

Примеры эксплуатации:

Пример доступа к информации о проекте по его id от имени авторизованного пользователя, которому не доступен список проектов:

```
GET /og/project/view?id=32724 HTTP/1.1
Host: pos.gosuslugi.ru
Cookie: <COOKIE>
```

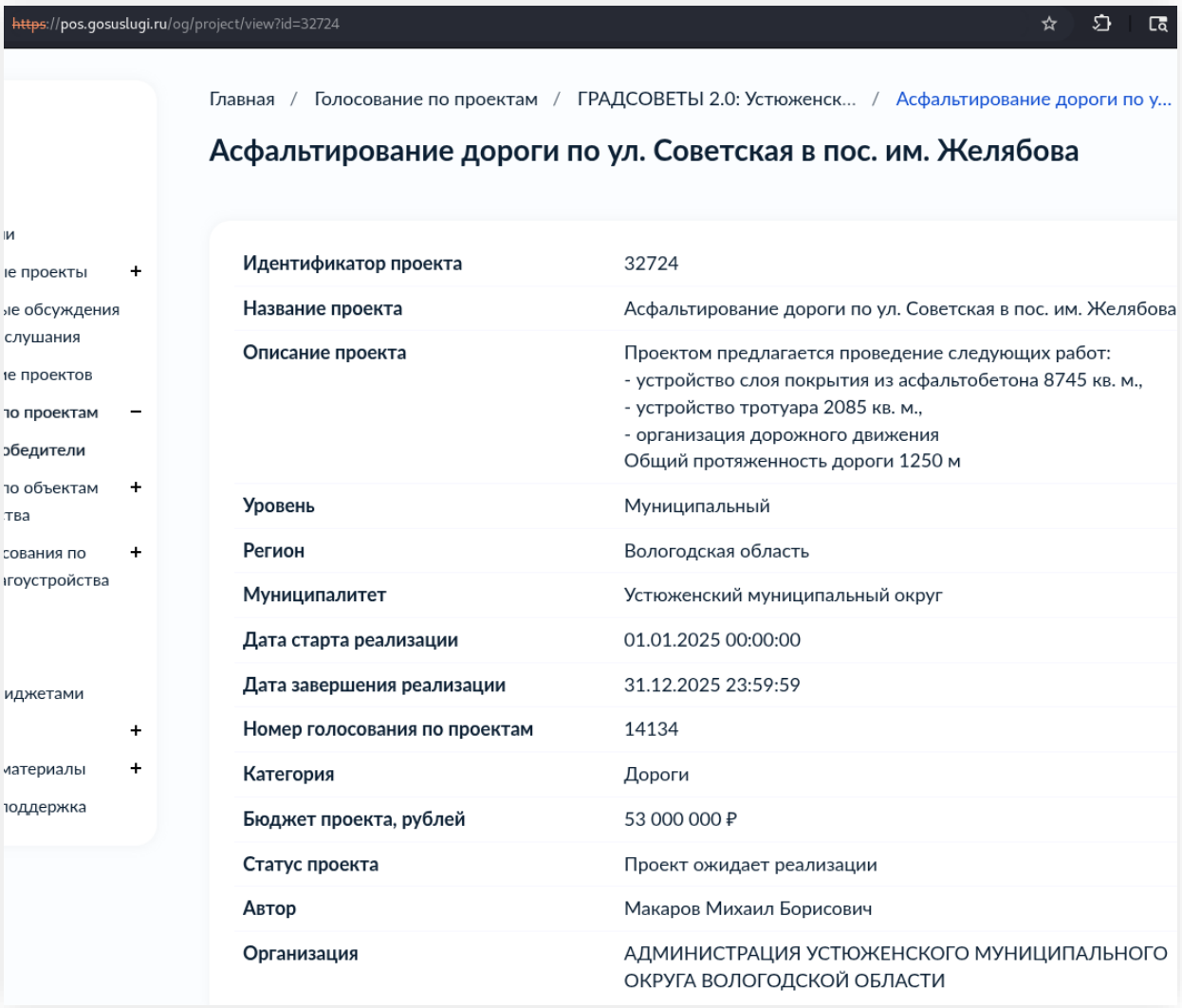

Рисунок 10. Просмотр информации о проекте (1)

Пример доступа к информации о проекте по его id от имени авторизованного пользователя, которому не доступен список проектов из раздела «Проекты-победители»:

```
GET /og/project/view-winner?id=26049 HTTP/1.1
Host: pos.gosuslugi.ru
Cookie: <COOKIE>
```

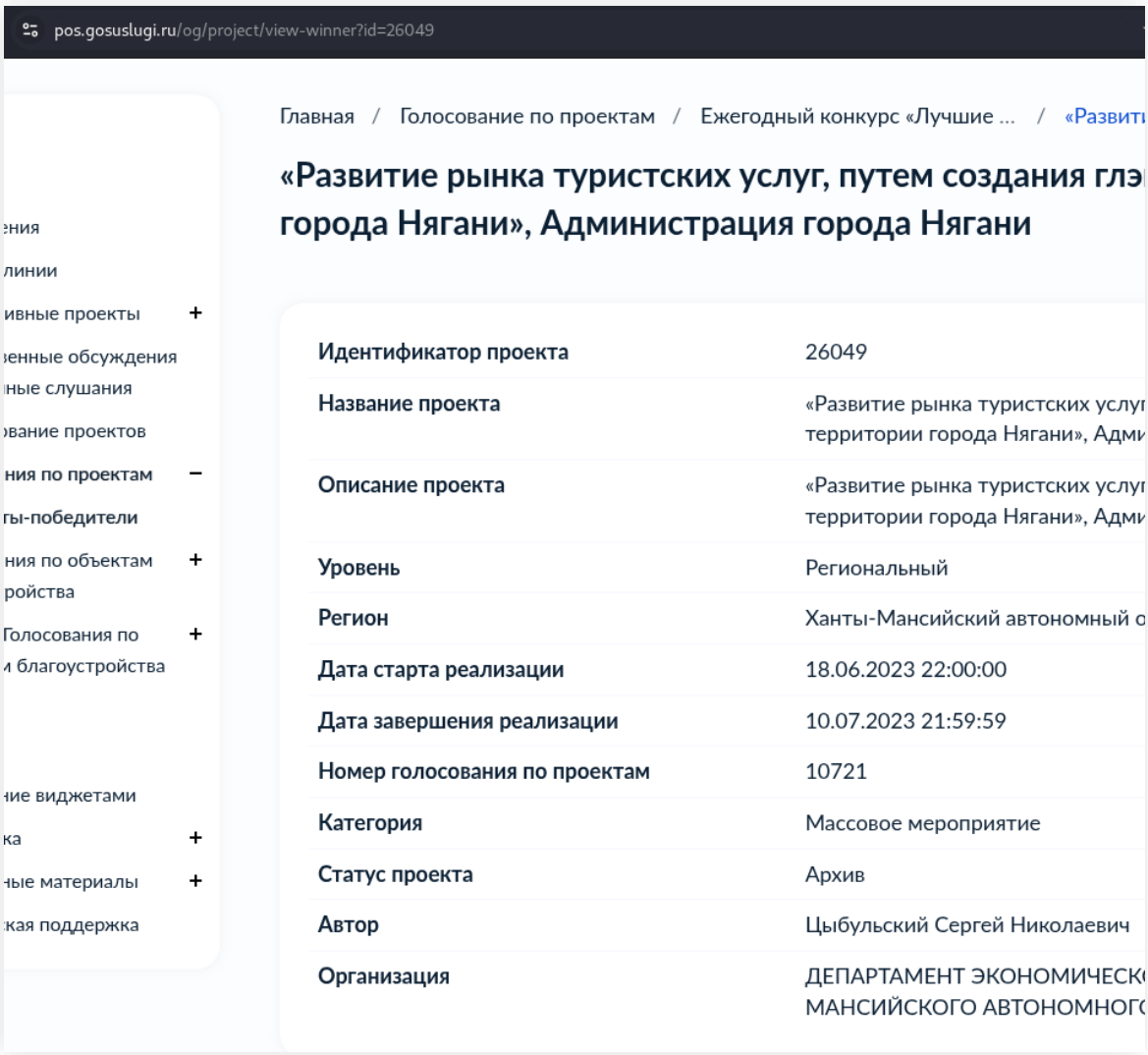

Рисунок 11. Просмотр информации о проекте (2)

Пример доступа к информации о проекте по его id от имени авторизованного пользователя, которому не доступен список инициативных проектов:

```
GET /og/initiative-budgeting/view?id=4 HTTP/1.1
Host: pos.gosuslugi.ru
Cookie: <COOKIE>
```

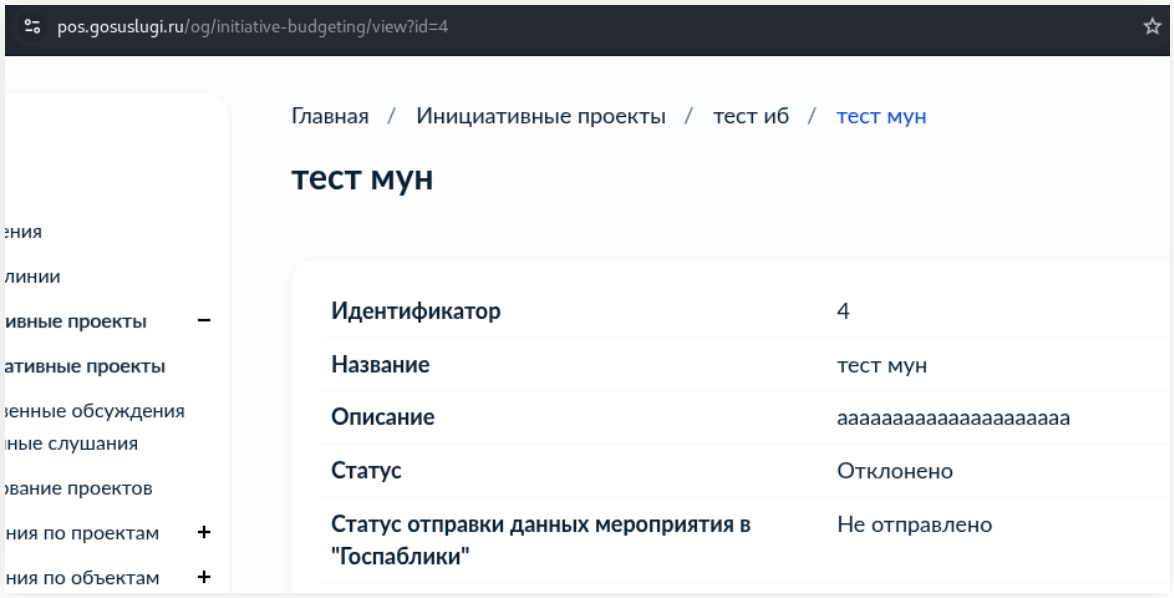

Рисунок 12. Просмотр информации о проекте (3)

Пример получения загруженных в приложение файлов от имени авторизованного пользователя с помощью сценария **/og/file-info/download**:

```
GET /og/file-info/download?file_info_id=773999 HTTP/1.1
Host: pos.gosuslugi.ru
Cookie: <COOKIE>
```

Пример получения загруженных в приложение файлов от имени авторизованного пользователя с помощью сценария **/og/document/file**:

```
GET /og/document/file?file_info_id=773999 HTTP/1.1
Host: pos.gosuslugi.ru
Cookie: <COOKIE>
```

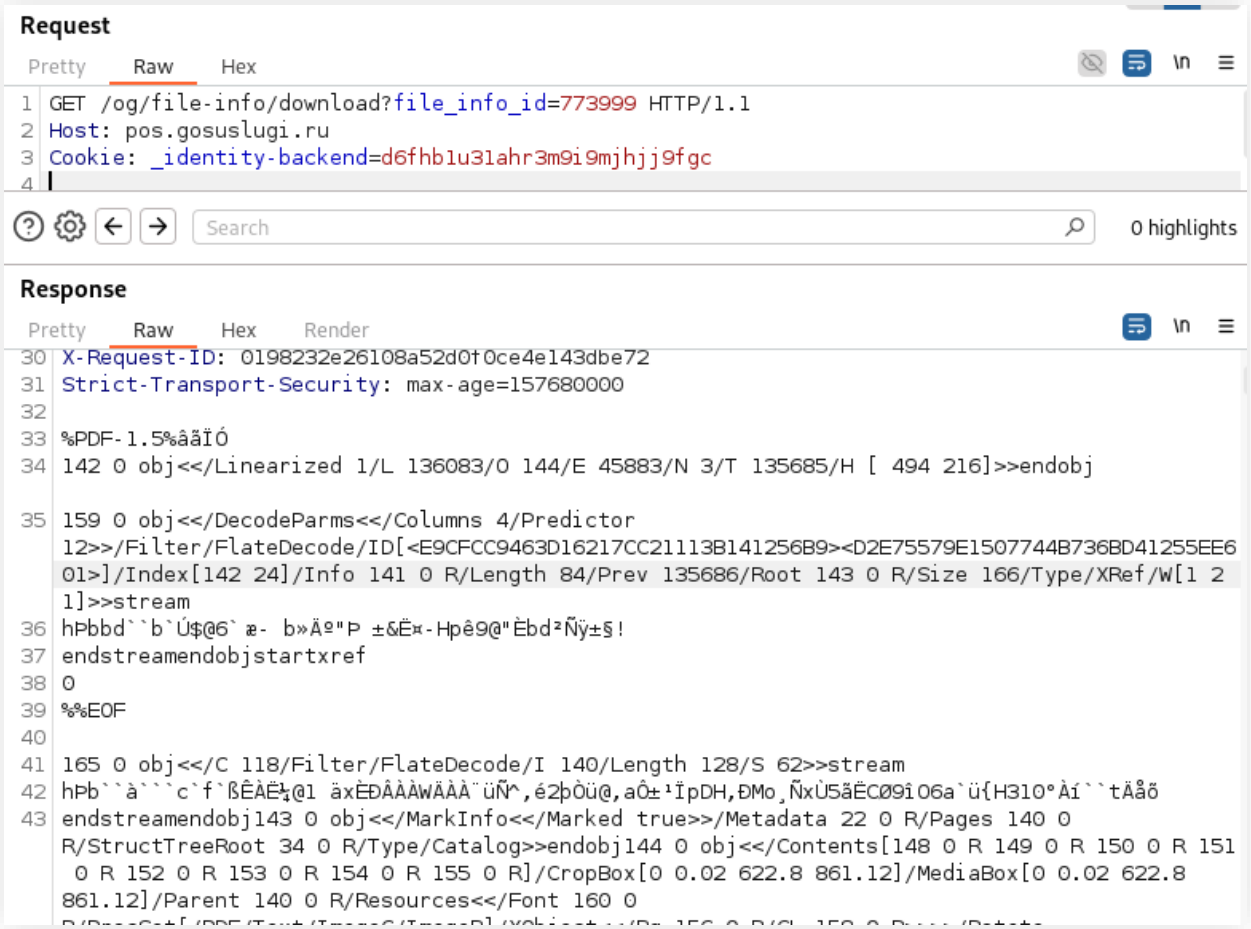

Рисунок 13. Успешное получение файла в сценарии /og/file-info/download

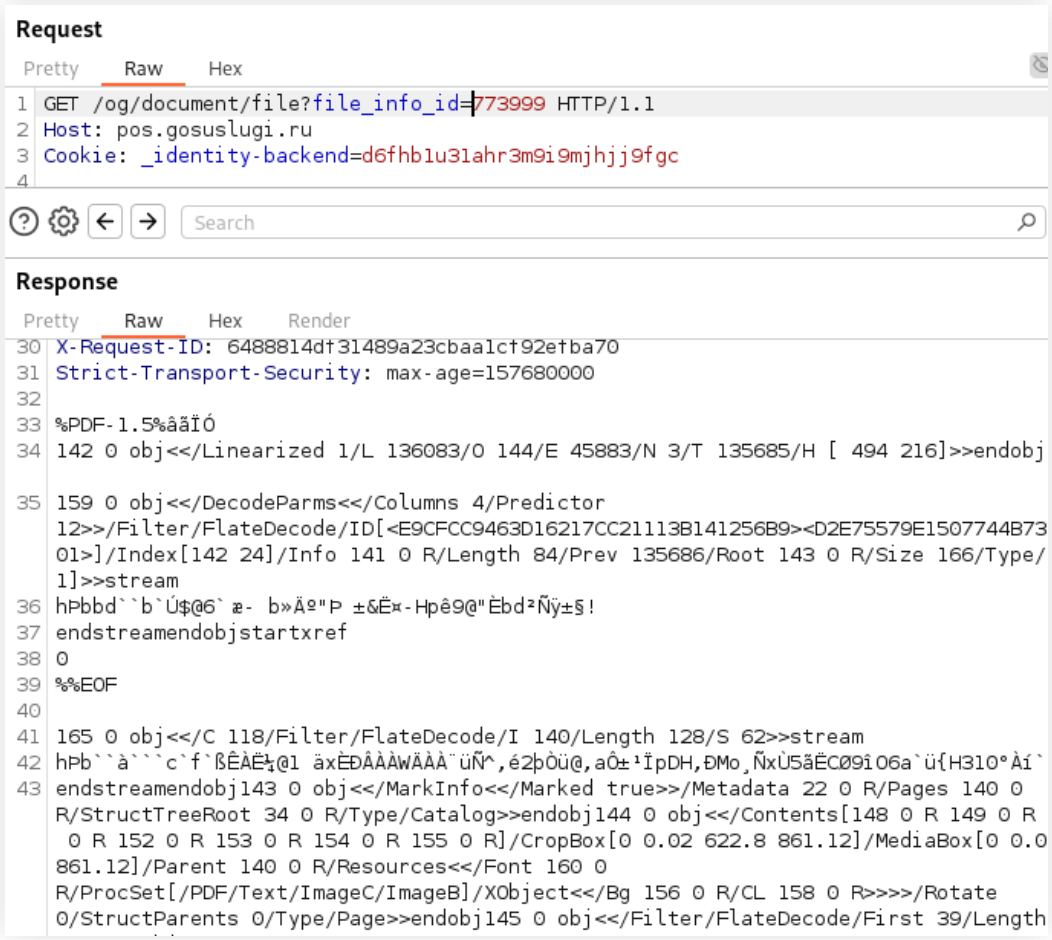

Рисунок 14. Успешное получение файла в сценарии /og/document/file

Рекомендации:

- Реализовать эффективное разграничение доступа к ресурсам:
  - ограничить доступ к информации или функциональности при прямом обращении к страницам и объектам приложения;
  - исключить кэширование страниц, содержащих конфиденциальную информацию;
  - разрешать доступ к защищенным ресурсам только после прохождения процедуры аутентификации.
- Проверять на стороне сервера привилегии пользователя до предоставления ему доступа к данным и функциональности приложения.
- Использовать устойчивые к атаке полного перебора идентификаторы объектов, например GUID.
- Рассмотреть возможность внедрения и использования библиотек и фреймворков, направленных на управление аутентификацией и авторизацией пользователей.
- Более подробные рекомендации по безопасной настройке контроля доступа можно найти по адресу: https://cheatsheetseries.owasp.org/cheatsheets/Authorization_Cheat_Sheet.html.

3.4. Межсайтовый скриптинг отраженный (Reflected XSS)

Статус: ***Не исправлено***
Критичность: ***Средняя***
Вероятность эксплуатации: ***Средняя***
Итоговый риск: ***Средний***
Модель нарушителя: [H2]

Результат перепроверки:

Уязвимость воспроизводится без изменений.

Описание:

в контролируемых пользователями данных некорректно фильтруются или кодируются специальные символы HTML и JavaScript. Таким образом, передача специально сформированной нагрузки приводит к тому, что пользовательские данные интерпретируются браузером как часть исходного кода страницы веб-приложения. В результате потенциальный злоумышленник может внедрить произвольный JavaScript-сценарий на страницу веб-приложения, который будет выполнен в браузере пользователя при ее посещении. Успешное проведение атаки «Межсайтовое внедрение сценариев» может привести к компрометации информации пользователей и/или выполнению произвольных действий на страницах веб-приложения. При этом все действия будут выполнены от имени и с привилегиями пользователя, посещающего страницу.

Риск:

- получение доступа к чувствительной информации;
- изменение содержимого отображаемой страницы;
- выполнение действий от имени пользователей приложения;
- компрометация сессионных данных пользователей.

Технические детали:

Уязвимые хосты:

- pos.gosuslugi.ru

В ходе работ были обнаружены уязвимости, позволяющие проводить атаки «Межсайтовое внедрение сценариев», подробная информация о которых приведена в таблице ниже (см. Таблица 5).

Таблица 5. Сведения об обнаруженных уязвимостях, позволяющих проводить атаки «Межсайтовое внедрение сценариев»

Сценарии	Параметры	Описание
/og/widgets/preview-code	code	В приложении в разделе управления виджетами имеется возможность настраивать HTML-код виджета. Авторизованный пользователь может просматривать превью, при этом приложение обрабатывает параметр code,

Сценарии	Параметры	Описание
		содержащий HTML-код виджета в кодировке Base64. Это позволяет помимо HTML-кода виджета внедрить JavaScript-код и закодировать его в Base64, что приводит к отраженному межсайтовому скриптингу (Reflected XSS).
/og/news-minstroy/create		В приложении в функциональности загрузки изображений при прикреплении файла с названием, в котором содержится код ***"><input autofocus onfocus=alert(document.cookie)>.png***, будет выполнен JavaScript-сценарий.

Пример эксплуатации:

Пример атаки в сценарии **/og/widgets/preview-code**:

```
GET /og/widgets/preview-
code?code=PGRpdiBpZD0iZTMyOWZiNDAtd2lkZ2V0LXBvcyIgc3R5bGU9IndpZHRoOiAzMjBweDsgaGVpZ2h0OiAzMjBweDsiPjwvZGl2Pgo8c2NyaXB0IGlkPSJlMzI5ZmI0MCIgc3JjPSJodHRwczovL3Bvcy5nb3N1c2x1Z2kucnUvb2cvd2lkZ2V0L2pzL21haW4uanMiIGRhdGEtc3JjLWhvc3Q9Imh0dHBzOi8vcG9zLmdvc3VzbHVnaS5ydS9vZyIgZGF0YS1vcmctaWQ9IjI4MDY3Ij48L3NjcmlwdD4= HTTP/1.1
Host: pos.gosuslugi.ru
Cookie: <COOKIE>
```

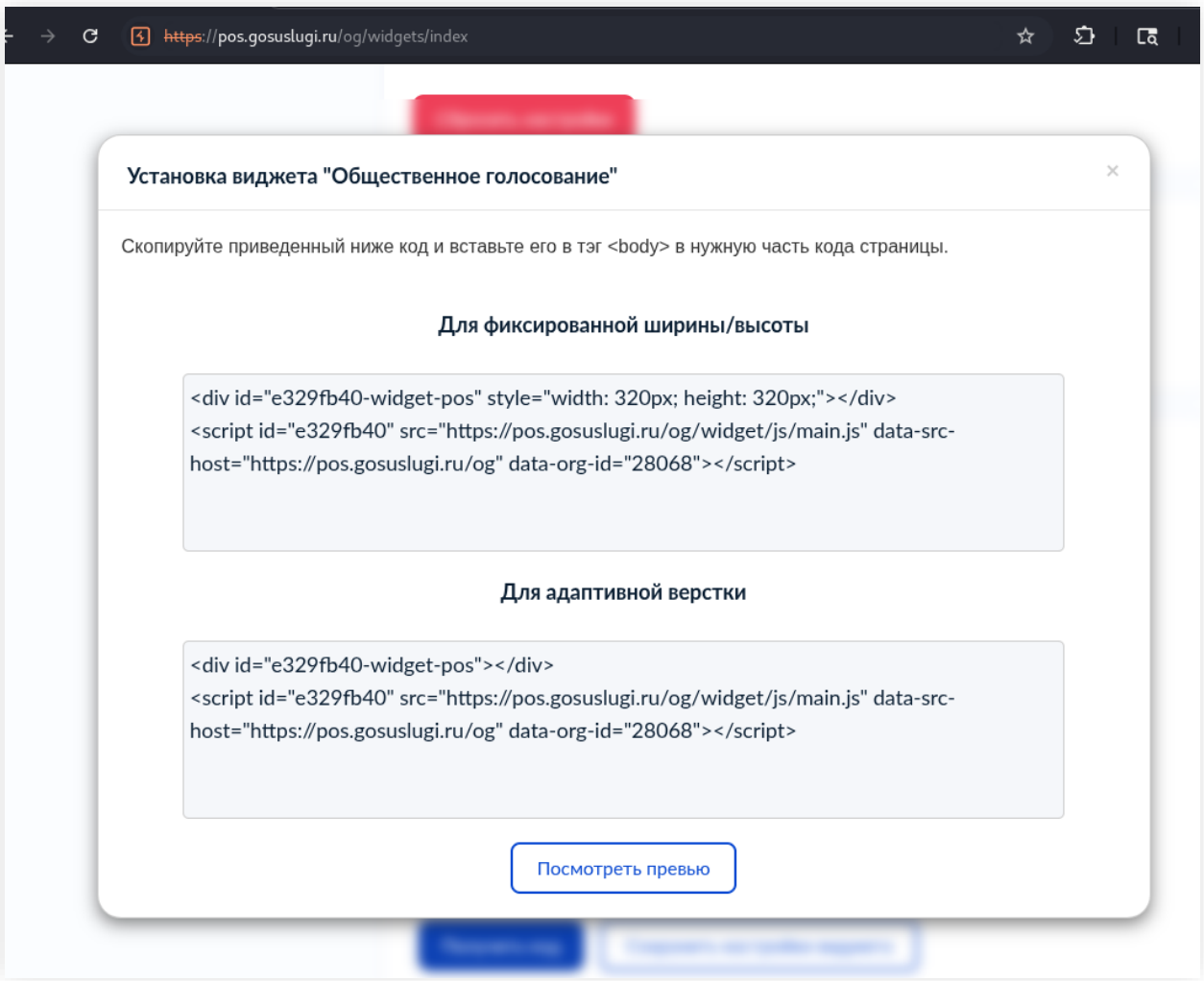

Рисунок 15. Пример настраиваемого виджета

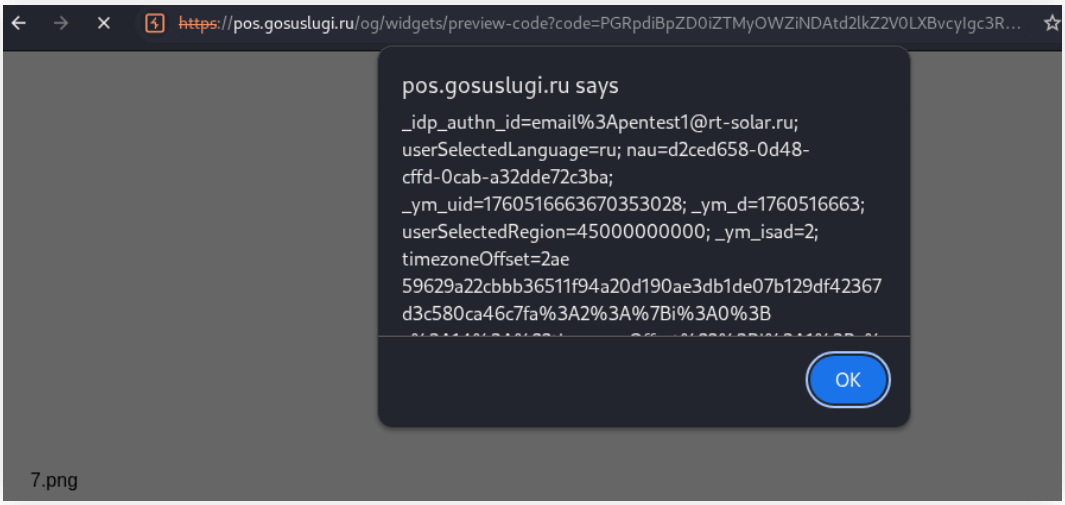

Рисунок 16. Пример выполнения JavaScript-сценария при просмотре виджета

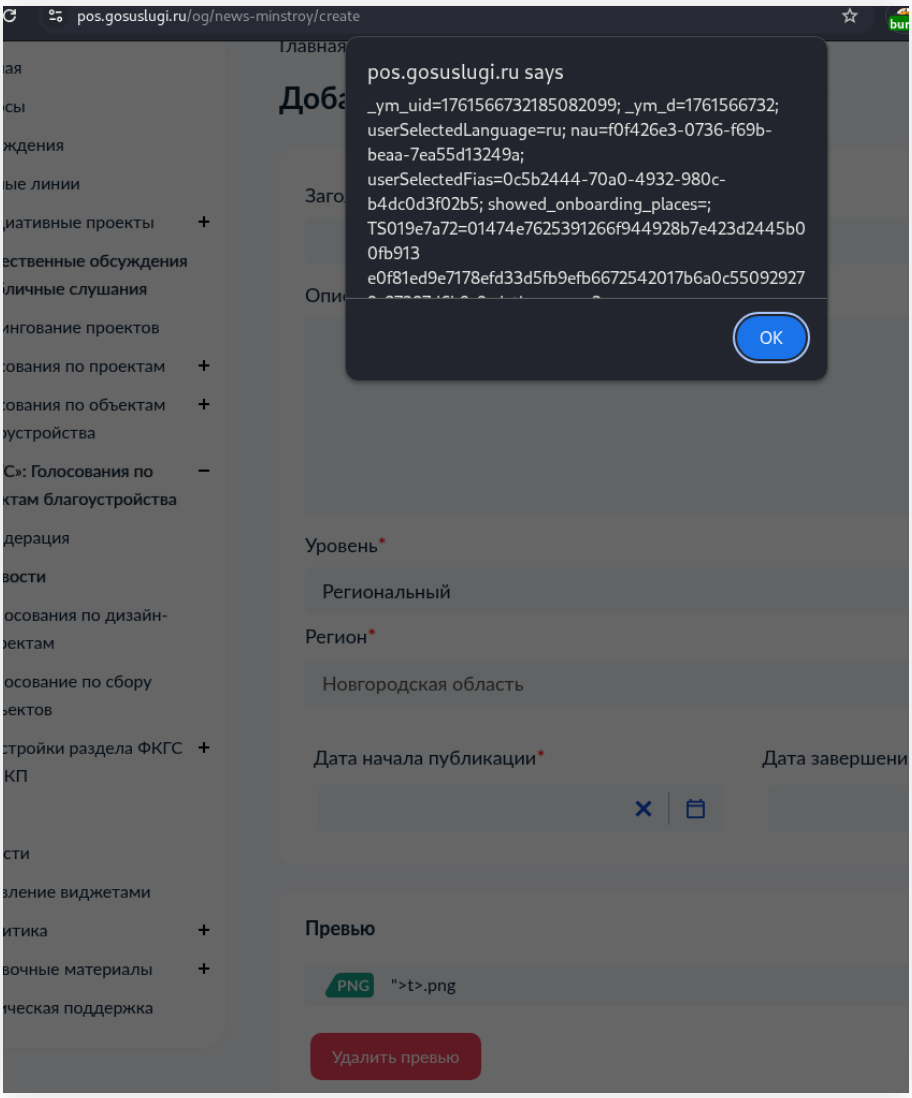

Рисунок 17. Пример загрузки изображения и выполнения JavaScript-сценария

Рекомендации:

- Корректно обрабатывать контролируемые пользователем данные, в том числе HTTP-заголовки запросов:
  - не встраивать напрямую на страницу данные, полученные из недоверенных источников;
  - заменять потенциально небезопасные символы в синтаксисе HTML на их эквиваленты, которые не являются символами форматирования.
  - проверять данные как на стороне клиента, так и на стороне сервера.
- Не использовать потенциально опасные JavaScript-функции и DOM-объекты, которыми может воспользоваться потенциальный злоумышленник для проведения атак.
- Рассмотреть возможность внедрения межсетевого экрана уровня приложений (Web Application Firewall).
- Использовать фреймворки, которые автоматически обрабатывают данные пользователей для защиты от проведения атак типа «Межсайтовое выполнение сценариев».
- Включить и корректно настроить политику защиты контента (Content Security Policy).
- Устанавливать для cookie-файлов с сессионными данными атрибуты Secure и HttpOnly.
- Более подробные рекомендации по защите от атак «Межсайтовая подделка запросов» можно найти по адресу: https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html.

3.5. Возможность проведения атаки «Подделка запроса на стороне сервера (SSRF)»

Статус: ***Исправлено***
Критичность: ***Средняя***
Вероятность эксплуатации: ***Средняя***
Итоговый риск: ***Средний***
Модель нарушителя: [H1]

Описание:

при работе приложение выполняет некоторые запросы к внутренним или внешним сервисам, используя контролируемые пользователем данные: URL-адрес, домен, сценарий или прочие данные. При этом на стороне сервера не установлен список разрешенных для обращения ресурсов и не проверяется, к какому именно ресурсу отправляется запрос. В результате при обработке пользовательских данных сервер может совершать обращения к произвольным ресурсам: как к внешним, так и к внутренним. Таким образом, потенциальный злоумышленник может отправлять от имени сервера запросы для получения доступа к чувствительным данным или сведениям о структуре внутренней сети.

Риск:

- сканирование внутренней сети;

- получение доступа ко внутренним системам;
- обход средств контроля доступа;
- использование сервера для проведения атак типа «отказ в обслуживании» на другие системы и сервисы;
- компрометация чувствительных данных, связанных с пользователями, или обрабатываемой в приложении информации.

Технические детали:

Уязвимые хосты:

- pos.gosuslugi.ru

Уязвимость обнаружена в форме голосования «Подача жалобы о вредоносном ресурсе».

Сценарий: `/lkp/poll-anti-fishing/process/`

Уязвимый параметр: `"questionId":11`

```
POST /lkp/poll-anti-fishing/process/ HTTP/1.1
Host: pos.gosuslugi.ru
Cookie: <COOKIE>
Content-Type: application/json
Connection: keep-alive

{"isLocal":true,"_csrf":"UjTtiWH40spaj9EDky8OhEdcmBB-OW-vQO1ykrb-O68qR6HPKqqA-yDHlEn-TGXUNh7hfCtePuUk2yP9xbFJxA==","answers":[{"questionId":10,"answerId":null,"answerValue":"http://<RANDOM_URL>"},{"questionId":11,"answerId":null,"answerValue":"http://<SSRF_URL>"},{"questionId":12,"answerId":null,"answerValue":"2025-11-01"},{"questionId":13,"answerId":6,"answerValue":null,"problem":null,"solution":null},{"questionId":14,"answerId":null,"answerValue":"<EMAIL>"}]}
```

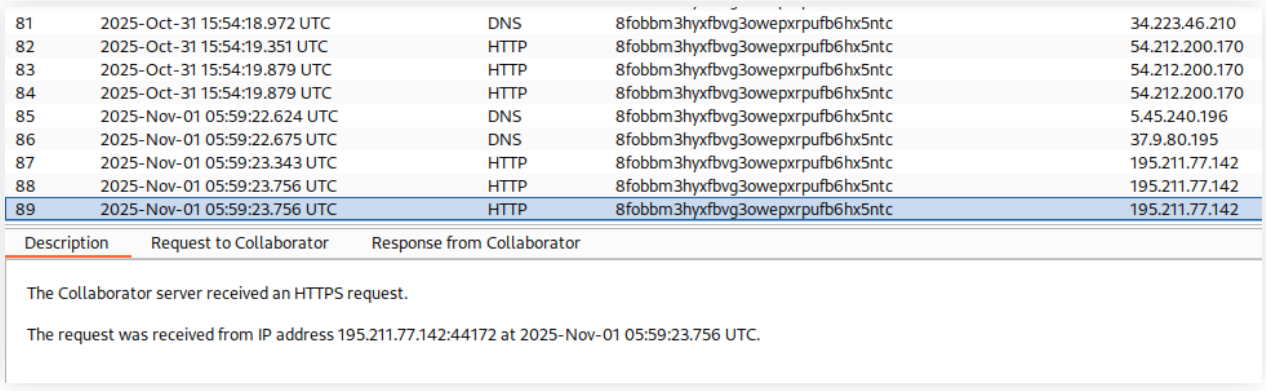

Рисунок 18. Множественные запросы после отправки формы

Рекомендации:

- В случае, если необходимо обращение к определенным доверенным ресурсам: проверять получаемые от пользователей данные по белым спискам адресов ресурсов, к которым разрешено обращение.
- В случае, если необходимо обращение к произвольным внешним ресурсам: проверять получаемые от пользователя данные для исключения обращения к каким-либо ресурсам внутренней сети.
- Установить список разрешенных сетевых маршрутов для ограничения доступа к произвольным ресурсам.
- Более подробные рекомендации по защите от атаки «Подделка запроса на стороне сервера (SSRF)» можно найти по адресу: https://cheatsheetseries.owasp.org/cheatsheets/Server_Side_Request_Forgery_Prevention_Cheat_Sheet.html.

3.6. Небезопасная конфигурация механизма кэширования

Статус: ***Не исправлено***
Критичность: ***Средняя***
Вероятность эксплуатации: ***Низкая***
Итоговый риск: ***Низкий***
Модель нарушителя: [H1]

Результат перепроверки:

Уязвимость воспроизводится без изменений.

Описание:

кэширование позволяет увеличить производительность и уменьшить время ответа сервера, поскольку предполагает повторное использование ранее извлеченных данных. В отличие от базы данных, в кэше информация хранится ограниченное время, после чего – удаляется. В случае наличия в кэше ответа на полученный запрос сервер вернет его без повторной серверной обработки.

Вследствие некорректной настройки сервера в кэше сохраняются чувствительные данные пользователей. В результате в течение времени жизни кэша потенциальный злоумышленник может получить доступ к чувствительным данным пользователей и использовать их для проведения дальнейших атак.

Риск:

- изменение содержимого отображаемой страницы;
- раскрытие чувствительных данных пользователей;
- проведение атак на пользователей приложения.

Технические детали:

Уязвимые хосты:

- pos.gosuslugi.ru

Сценарий: `/og/api/v1/*`

Пример эксплуатации:

Пример запроса, кэширующего данные:

```
GET /og/api/v1/project?sort=-id&page=2&per-page=5 HTTP/1.1
Host: pos.gosuslugi.ru
Authorization: Bearer <TOKEN>
```

Стоит отметить, что при первоначальном запросе к API в веб-приложении требуется авторизационный токен. Далее получение сведений о проектах возможно без авторизационного токена, при сохранении полного пути и параметров запроса.

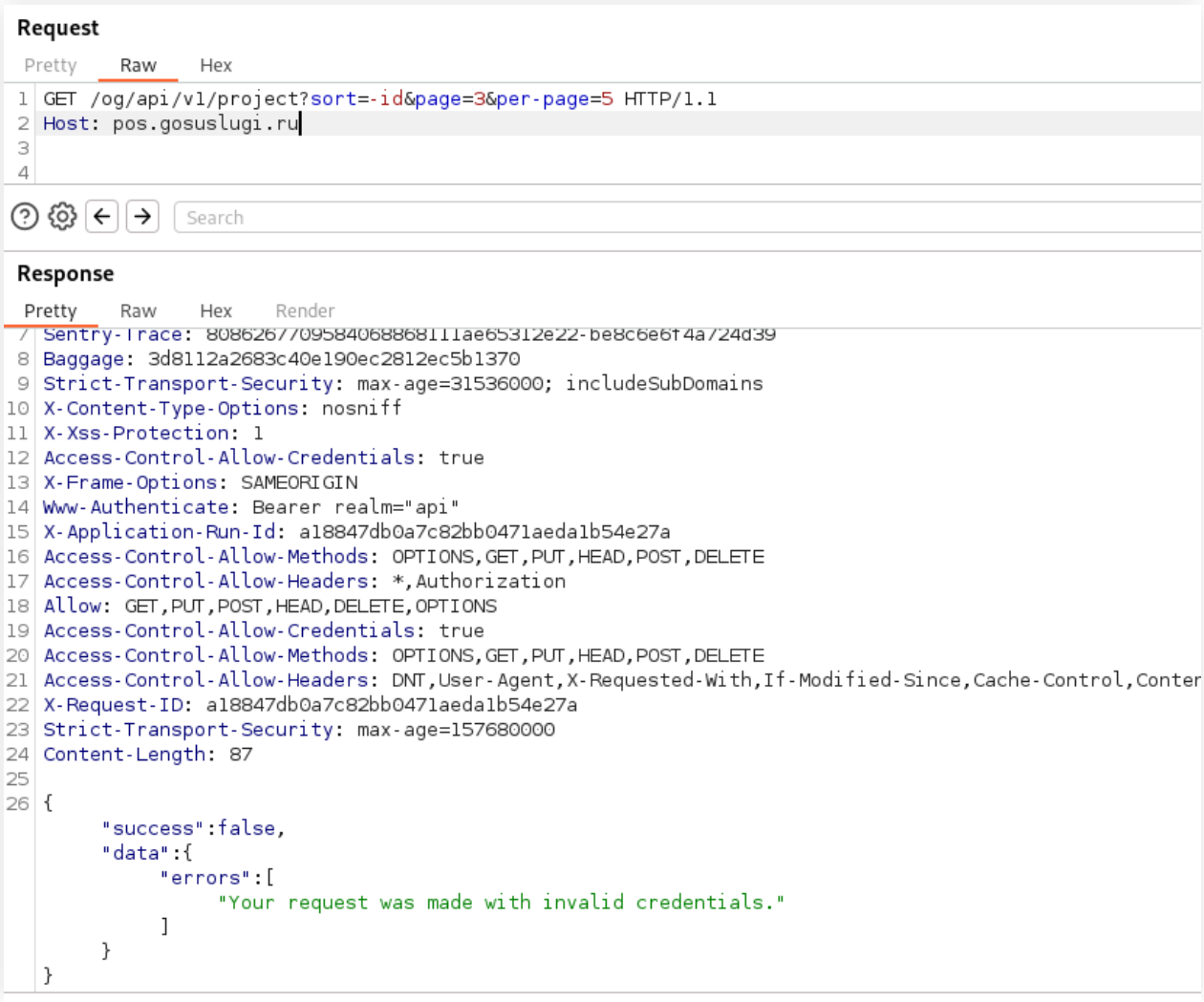

Рисунок 19. Отсутствие возможности получить сведения без авторизационного токена

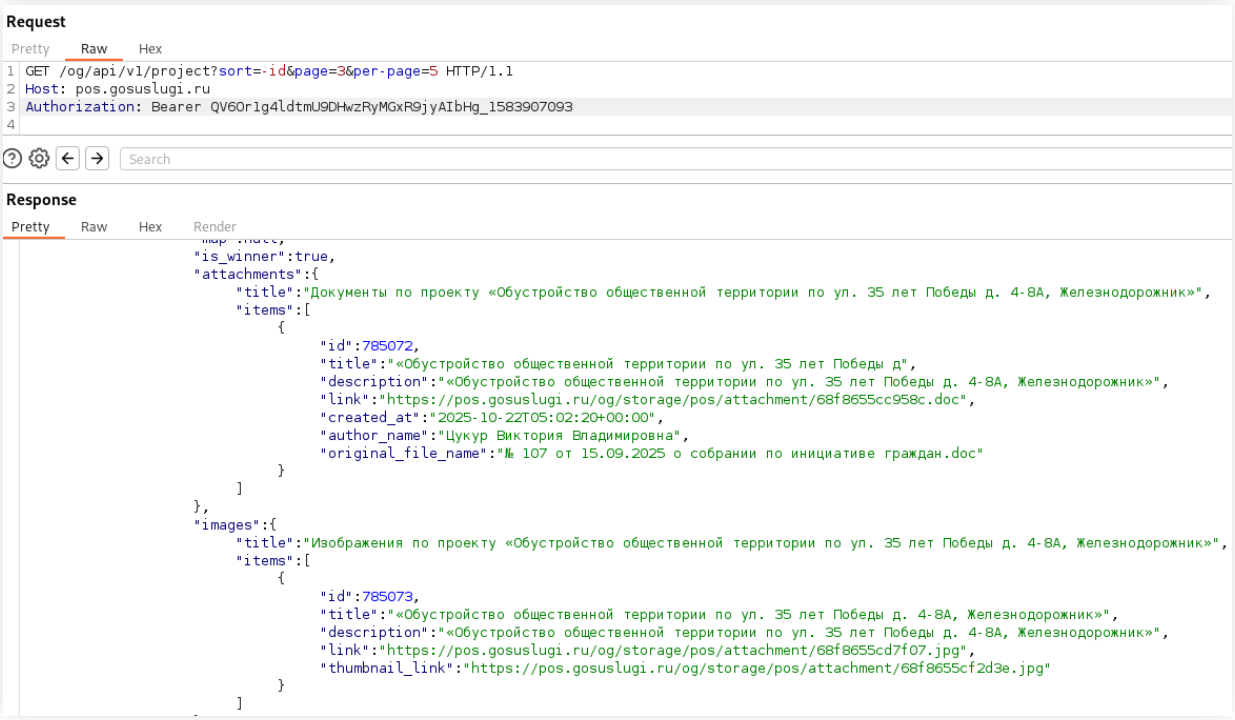

Рисунок 20. Получение сведений о проекте с токеном

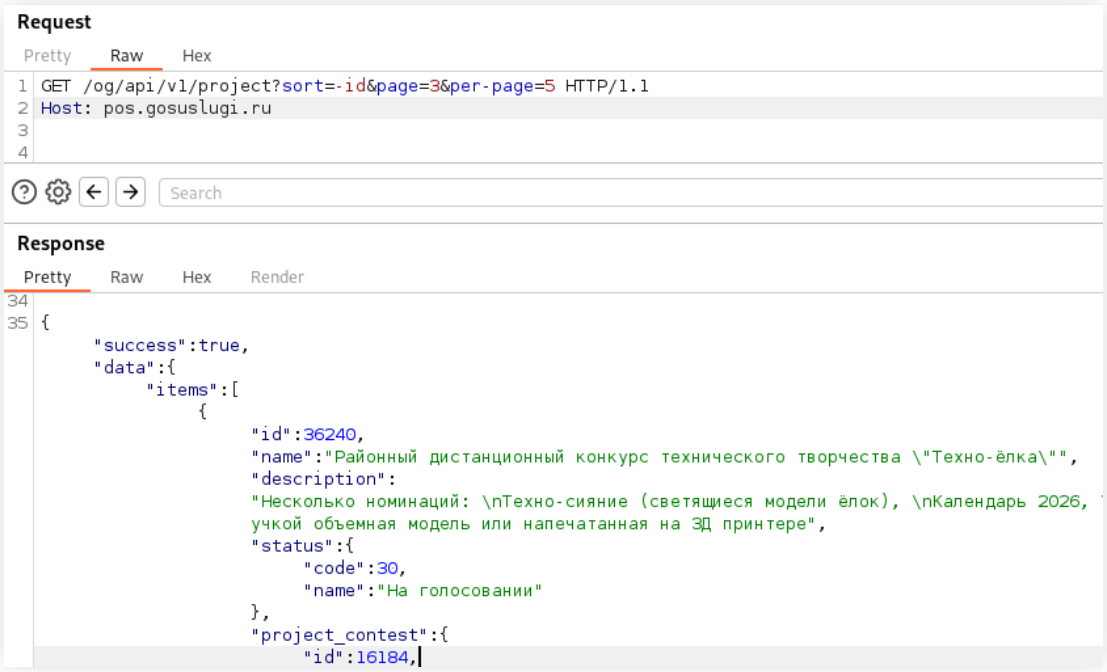

Рисунок 21. Получение сведений о проекте без авторизационного токена после первоначального запроса с токеном

Рекомендации:

- Корректно настроить механизмы кэширования данных или рассмотреть возможность их отключения.

3.7. Недостатки бизнес-логики приложения

Статус: ***Исправлено***
Критичность: ***Низкая***
Вероятность эксплуатации: ***Средняя***
Итоговый риск: ***Низкий***
Модель нарушителя: [H2]

Описание:

одна из основных целей бизнес-логики приложения – обеспечить соблюдение пользователями правил и ограничений, которые были установлены при разработке приложения, т. е. бизнес-логика должна определять поведение приложения при всех возможных сценариях использования. Однако недостатки в архитектуре или реализации приложения приводят к тому, что пользователь может взаимодействовать с приложением по непредусмотренному разработчиками сценарию.

В приложении реализована функциональность выгрузки деперсонализированной статистики для пользователей с доступом к функциональности **/og**. При этом если в проекте голосовал пользователь с учетной записью организации, то информация о нем будет включена в статистику без деперсонализации. Также любой пользователь с доступом к **/og** может получить деперсонализированную статистику любого голосования по ссылке.

Риск:

- обход установленных в приложении требований;
- получение доступа к чувствительной информации.

Технические детали:

Уязвимые хосты:

- pos.gosuslugi.ru

Сценарий: `/og/backend/api/v1/poll/download-anonymous-stats/<ID>`

Уязвимый параметр: `job_id`

Запрос на выгрузку деперсонализированной статистики голосования, доступ к которому отсутствует:

```
GET /og/backend/api/v1/poll/download-anonymous-stats/<ID> HTTP/1.1
Host: pos.gosuslugi.ru
Cookie: _identity-backend=<COOKIE>
```

В результате выполнения указанного запроса будет получено значение **job_id**.

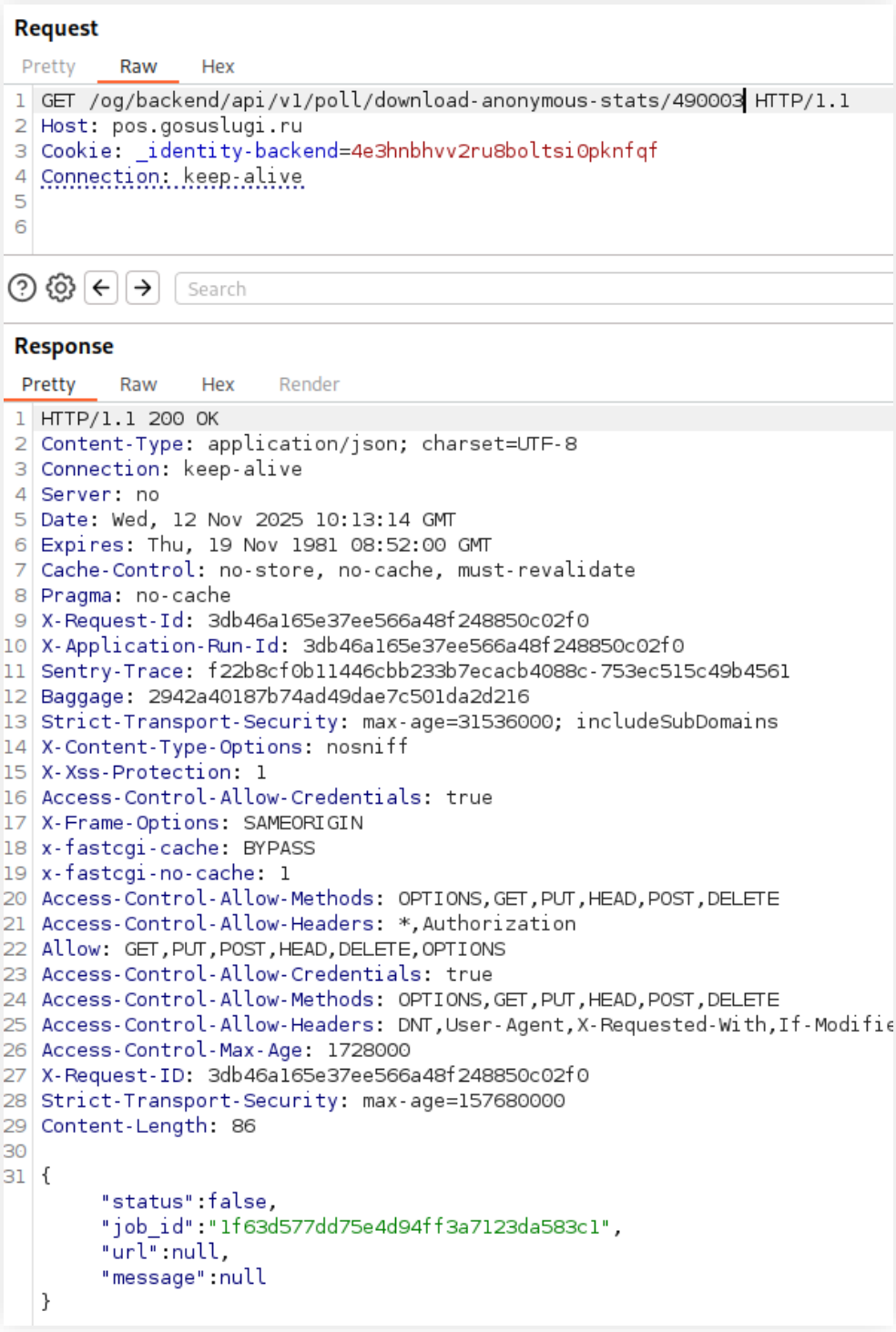

Рисунок 22. Получение значения идентификатора job_id

После получения идентификатора **job_id** необходимо сделать запрос для получения ссылки на отчет:

```
GET /og/backend/api/v1/poll/download-anonymous-
stats/490003?job_id=7cffb48cc3ed7f87c0daeff639f12073 HTTP/1.1
Host: pos.gosuslugi.ru
Cookie: _identity-backend=<COOKIE>
```

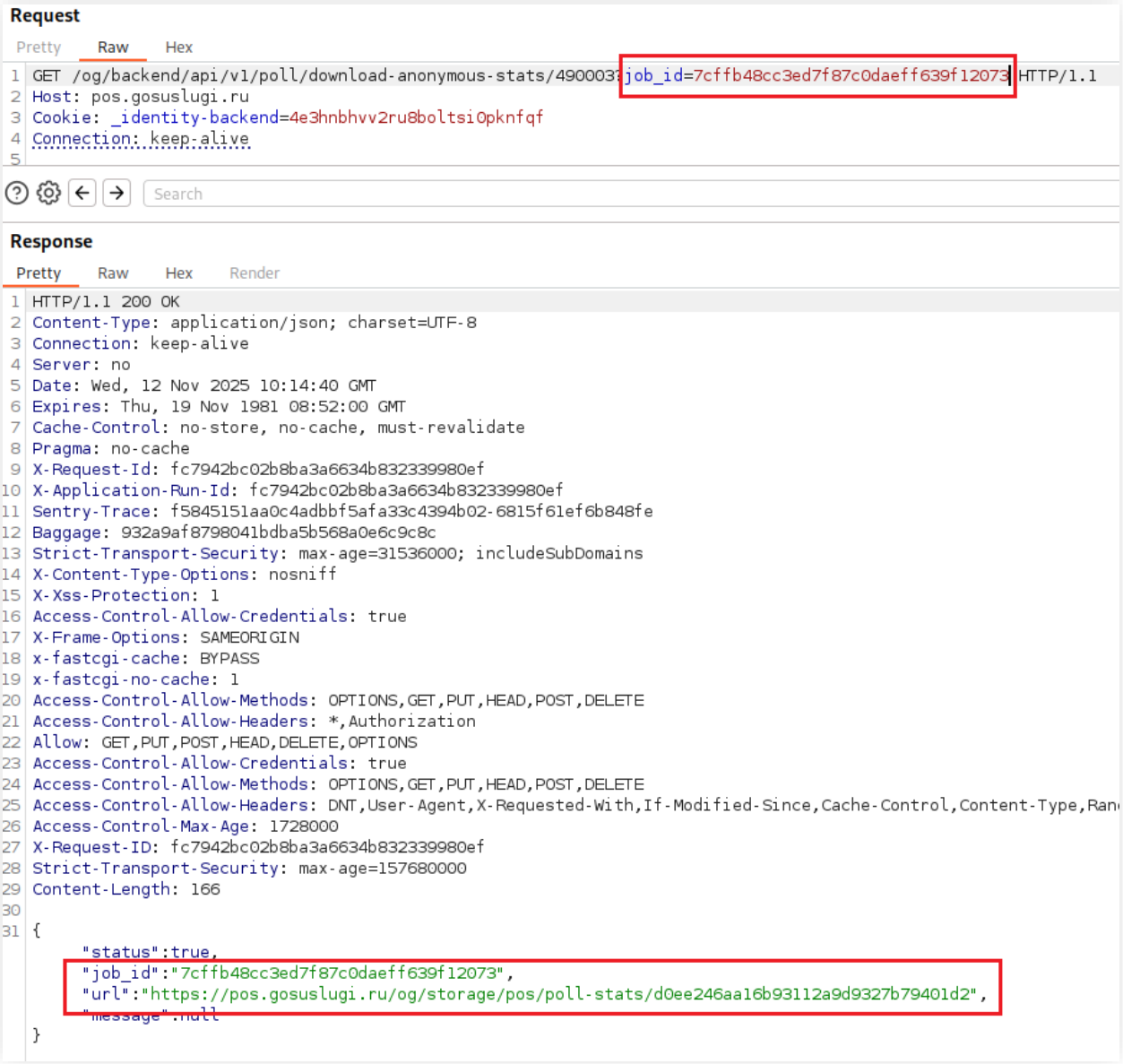

Рисунок 23. Получение ссылки для скачивания статистики

Request

Pretty Raw Hex

```
1 GET /og/storage/pos/poll-stats/d0ee246aa16b93112a9d9327b79401d2 HTTP/1.1
2 Host: pos.gosuslugi.ru
3 Connection: keep-alive
4
```

Search

Response

Pretty Raw Hex Render

```
1 HTTP/1.1 200 OK
2 Content-Type: text/plain; charset=utf-8
3 Content-Length: 14881
4 Connection: keep-alive
5 Server: no
6 Date: Wed, 12 Nov 2025 10:18:02 GMT
7 Content-Disposition: attachment; filename=poll_report_490003
8 Content-Security-Policy: block-all-mixed-content
9 ETag: "33b996d95d34ac1316a0c63e09fa650a"
10 Last-Modified: Wed, 12 Nov 2025 10:14:33 GMT
11 No-Gzip-Compression: true
12 Strict-Transport-Security: max-age=31536000; includeSubDomains
13 Vary: Origin
14 Vary: Accept-Encoding
15 X-Amz-Request-Id: 18773B077D060736
16 X-Content-Type-Options: nosniff
17 X-Xss-Protection: 1; mode=block
18 X-Cache-Status: MISS
19 X-Request-ID: 263b20e6adb39c8b9811b7c9b9459a7b
20 Accept-Ranges: bytes
21 Strict-Transport-Security: max-age=157680000
22
23 PKÐQl[fÞvHA[Content_Types].xmlµËNÃ0E÷ýÈ[»eJÚ%T¢|±'U¿äqKû÷LUU@ÐlëÎÌ=\Í¶ÎHh¯ÙY^mü²fo§
24 >Ïeîj°é¨ ¯zF®m.î±«_3£5JfT[Ön/~¼ñú«Ü#ñ¶ÁÖD¼:u!¿l^¨QÉhøOh£@µvÅl&[ì,ìWî¤ñó\¦ü,U[+>BZ½°
   wN*.ðß{Ý¨kÐ?" ÒJô/ÅôPKÐQl[àÍ;xl/workbook.xmlÁNÃ0ï}ÈwBU]ÒÎÀd»FKì*PoçmÙþüÿýî;'sÆ"íí¦4
25 ú 3bÍÉ>¶íÍ>¬®ÜÃàì#¾ðøê
26 )|Uõ2ÇEÀ5F£¿¾·fC>«ô·K½U;¼êLé¢enÃê«îfÄCB°î³¼¦·ÿÆÝ/PKÐQl[lÿ²àxl/styles.xmlÅTKKÄO¾ï¯¹k
   þÒqî hãÎûþWu\1wiz®ÁÓ«Õ¶ÄõºÚ$%I$7Dl¡qAõNÊ;Tö9NÀD&[A£®+uèIè9Ã¤ Æ"Ûns\É¸Y3ÅcØ#bkEOð&ý_
   ¸B°è¾Alð®F¡þ,ÂLb~Tð,lcx`à.`'¥÷Í$°¾CibQ{cù³Øéo±£~/E«ïÍ*ê(3^?oÕ**àvå¾9ßcö=N-oL[pä
   é9Ï'ÇÏáGÇæOÙÈäø7PKÐQl[PpÚ)æ]_rels/.rels-ÁJÄOïû!÷mêD¤é".ÂÞDV<É´m!ZßPA]ª`nIþùþf;ûI<bÊ
27 &Zz-oWë3¹mWÍNP8GYðLÈZ¥Ð¹RÙè!WOðMÂÛÔ+3BjS×§*}fÈv%xÅPjöö.¦ñ>ÆQÃ3áozb×9»h<òMÝ!õX´'õôVW
   MÁ
28 ÂODïý{Mõ "i ët«d7$[Ñ¿7 ¨syÃ0zxÆ 'ìåzÕIèhòxëåu<µ;)
29 [l ^¾ ÈÁ4ú)AfEÔ,½¼3§½RÅÝ!Ú²ª+)GËÕæ¢yöäEjÓu[O`jÓwPFTéCJÁ;Ëõ¹$ZX«ÿèSm}Óiõ3V¿æPKÐQl[òu
   à¶ìfo/ÿ_y¯¯ùú ÚhÖÉN&E EWKÝè½ÚÄ(rë·Á¡u¹Êa¢³ðj;ÖKpQiÇ)ÐÎ{Ãõvb»$:ÛÎ*îCil¸øâà{¬Áó{al#µX
   él|i|ÓFoYPÂÉñäB-Âª¶ò;îÙ2ÉÃr-jåð'Ê@«+< }ùPKÐQl[ñøXxDxl/_rels/workbook.xml.rels-ÏNÃ0¢
```

Рисунок 24. Скачивание статистики

A	B	C	D	E	F	G	H	I	J	K
Пол¶ождения		Возраст	Регион проживания пользователя¶живания пользователя¶гион адреса регистрации¶ипалитет адреса регистрации				Юридическое лицо	Название организации	Тип организации	ИНН организации
Пол¶ождения		Возраст	Регион проживания пользователя¶живания пользователя¶гион адреса регистрации¶ипалитет адреса регистрации				Юридическое лицо	Название организации	Тип организации	ИНН организации
Женский	¶989-04-1¶	36	Новгородская область	Великий Новгород	Новгородская область	Великий Новгород				
Мужской	¶986-08-1¶	39	Новгородская область	город Старая Русса	Новгородская область	город Старая Русса				
Женский	¶989-04-2¶	36	Новгородская область	Хвойнинский	Новгородская область	Хвойнинский				
Женский	¶991-05-2¶	34	Новгородская область	Великий Новгород	Новгородская область	Парфинское				
Мужской	¶986-02-1¶	39	Новгородская область	Савинское	Новгородская область	Савинское				
Мужской	¶991-01-0¶	34	Новгородская область	Ермолинское	Новгородская область	Ермолинское				
Женский	¶982-09-2¶	43	Новгородская область	Успенское	Новгородская область	Успенское				
Женский	¶980-11-2¶	44	Новгородская область	Великий Новгород	Новгородская область	Великий Новгород				
Мужской	¶988-08-0¶	37	Республика Бурятия	город Улан-Удэ	Республика Бурятия	город Улан-Удэ	Да	Тестовое общество	Юридическое лицо	1234567890
Мужской	¶988-08-0¶	37	Республика Бурятия	город Улан-Удэ	Республика Бурятия	город Улан-Удэ	Да	Тестовое общество	Юридическое лицо	1234567890

Рисунок 25. Пример содержимого статистики

Рекомендации:

- Изменить логику приложения таким образом, чтобы отсутствовала возможность обхода предусмотренной последовательности действий и условий.
- При разработке избегать неявных предположений о поведении пользователей или отдельных частей приложения.
- Ограничить в деперсонализированной статистике информацию о голосовавших организациях.

3.8. Небезопасные прямые ссылки на объекты, позволяющие получить данные уведомлений и документов

Статус: ***Не исправлено***
Критичность: ***Низкая***
Вероятность эксплуатации: ***Низкая***
Итоговый риск: ***Низкий***
Модель нарушителя: [H1]

Результат перепроверки:

Уязвимость воспроизводится без изменений.

Описание:

небезопасные прямые ссылки на объекты являются одним из типов недостатков контроля доступа. Взаимодействие с данными в приложении реализовано на основе контролируемого пользователем идентификатора. При этом в приложении отсутствует или некорректно реализована проверка наличия у пользователя прав доступа к запрашиваемым объектам. Таким образом, потенциальный злоумышленник может подобрать идентификаторы или использовать идентификаторы элементов, принадлежащих другим пользователям, для получения несанкционированного доступа к данным. Стоит отметить, что инкрементальные идентификаторы упрощают проведение атаки, поскольку являются предсказуемыми и легко подбираемыми.

Риск:

- получение доступа к данным других пользователей;
- чтение обрабатываемой в приложении информации;
- выполнение действий, недоступных пользователю из графического интерфейса.

Технические детали:

Уязвимые хосты:

- pos.gosuslugi.ru

В ходе работ был обнаружен ряд недостатков контроля доступа, позволяющих получить данные уведомлений и документов. Информация об указанных недостатках приведена в таблице ниже (см. Таблица 6).

Таблица 6. Обнаруженные недостатки контроля доступа

Сценарии	Параметры	Описание
/lkp/news/notification/?id=<ID>&type=notifications	id	Авторизованный пользователь может просматривать уведомления, предназначенные для других пользователей путем изменения идентификатора уведомления.
/og/docs/category/<ID>	–	В сценарии раскрывается документация и прочая информация, включая рейтинги регионов. Этот сценарий доступен любому пользователю.

Примеры эксплуатации:

Пример просмотра уведомления другого пользователя:

```
GET /lkp/news/notification/?id=2810065&type=notifications HTTP/1.1
Host: pos.gosuslugi.ru
Cookie: <COOKIE>
```

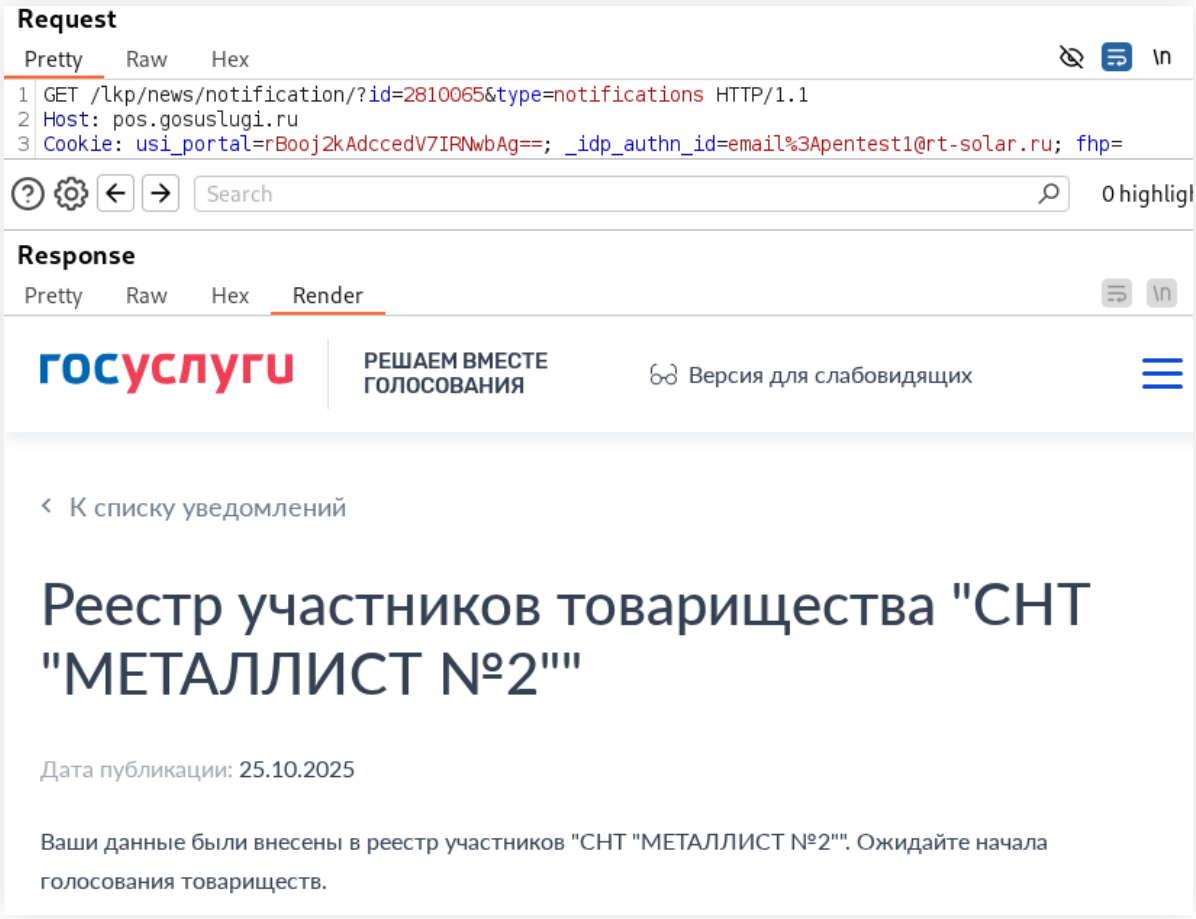

Рисунок 26. Просмотр уведомления другого пользователя

Пример получения данных о рейтинге регионов:

```
https://pos.gosuslugi.ru/og/docs/category/240
```

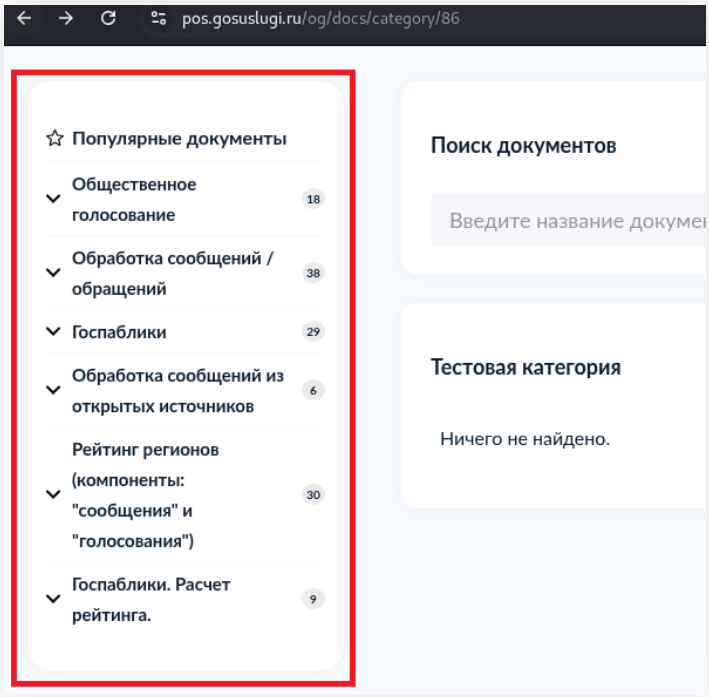

Рисунок 27. Пример получения доступа к данным

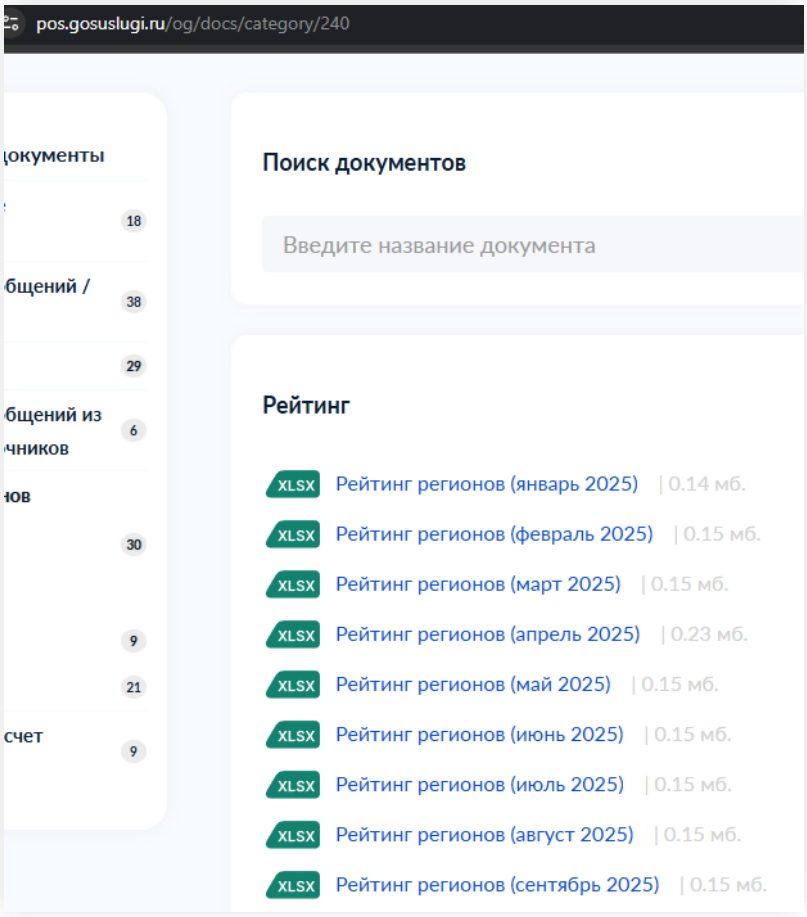

Рисунок 28. Пример получения доступа к данным о рейтинге регионов

Рекомендации:

- Реализовать эффективное разграничение доступа к ресурсам:
  - ограничить доступ к информации или функциональности при прямом обращении к страницам и объектам приложения;
  - исключить кэширование страниц, содержащих конфиденциальную информацию;
  - разрешать доступ к защищенным ресурсам только после прохождения процедуры аутентификации.
- Проверять на стороне сервера привилегии пользователя до предоставления ему доступа к данным и функциональности приложения.
- Использовать устойчивые к атаке полного перебора идентификаторы объектов, например GUID.
- Рассмотреть возможность внедрения и использования библиотек и фреймворков, направленных на управление аутентификацией и авторизацией пользователей.
- Более подробные рекомендации по безопасной настройке контроля доступа можно найти по адресу: https://cheatsheetseries.owasp.org/cheatsheets/Authorization_Cheat_Sheet.html.

3.9. Возможность проведения атаки «Внедрение HTML-кода на страницу веб-приложения»

Статус: ***Исправлено***
Критичность: ***Низкая***
Вероятность эксплуатации: ***Средняя***
Итоговый риск: ***Низкий***
Модель нарушителя: [H1]

Описание:

в контролируемых пользователями данных некорректно фильтруются или кодируются специальные HTML-символы. Таким образом, передача специально сформированной нагрузки приводит к тому, что пользовательские данные интерпретируются браузером как часть исходного кода страницы. В результате, потенциальный злоумышленник может внедрить произвольный HTML-код, например, ссылку на подконтрольный ресурс или форму ввода данных, и изменить отображение страницы для проведения атак на пользователей веб-приложения.

Риск:

- изменение содержимого отображаемой страницы;
- проведение атак на пользователей веб-приложения.

Технические детали:

Уязвимые хосты:

- pos.gosuslugi.ru

В приложении в функциональности поиска некорректно обрабатываются данные в двойных знаках `"`, в результате чего вводимый текст отображается как часть исходного кода страницы.

Сценарии:

- `/lkp/polls/`
- `/lkp/discussions/document`
- `/lkp/project-contests`
- `/lkp/rating`
- `/lkp/direct-line/list`

Уязвимые параметры:

- `plff[term]`
- `term`

Примеры эксплуатации:

```
https://pos.gosuslugi.ru/lkp/polls/?_pjax=%23polls_pjax&plff[category]=&plff[level]=10&plff[term]=1%22%26lt%3bHTML%26gt%3b%26lt%3bBODY%26gt%3ErandomText%22%3b&plff[type]=new

https://pos.gosuslugi.ru/lkp/direct-line/list/?dlf%5Blevel%5D=&dlf%5Blevel%5D=&dlf%5Bstatus%5D=&dlf%5Bterm%5D=11%22%26lt%3bHTML%26gt%3b%26lt%3bBODY%26gt%3ErandomText%22%3b
```

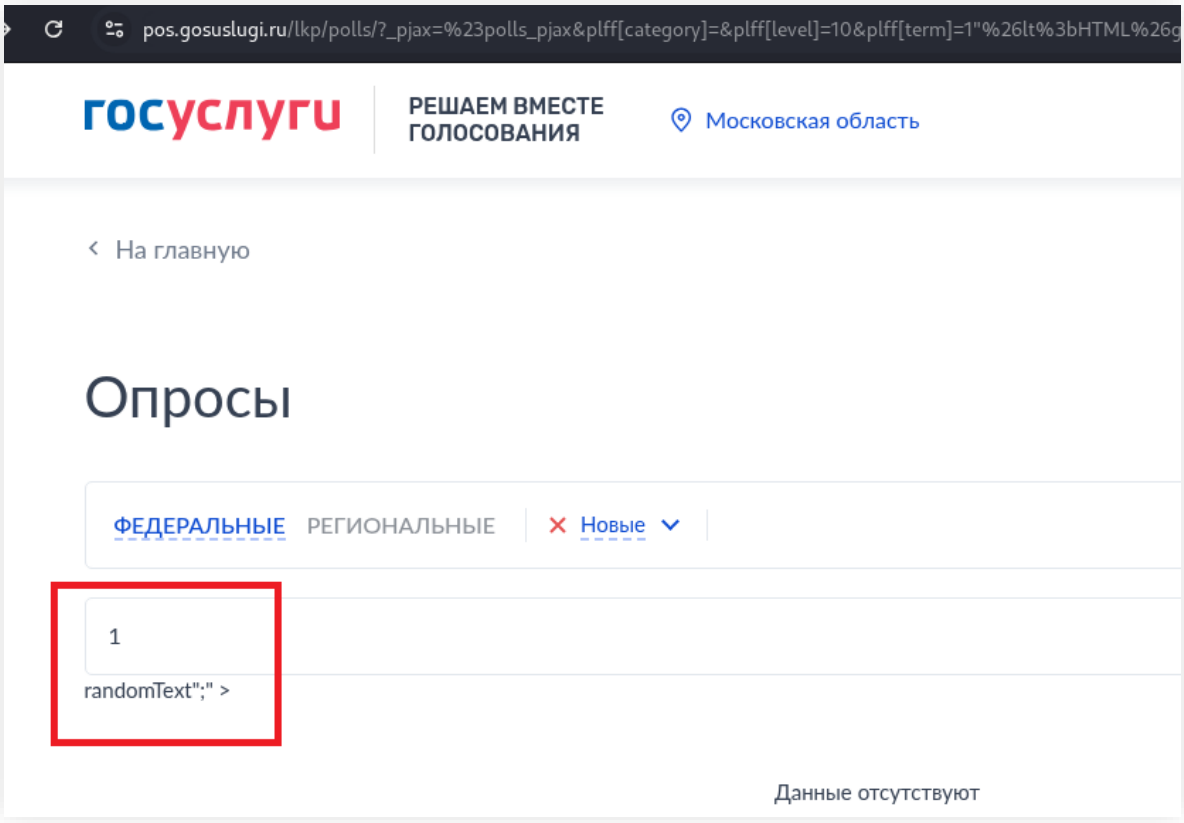

Рисунок 29. Пример отображения вводимых данных как части страницы

Рекомендации:

- Корректно обрабатывать контролируемые пользователем данные, в том числе HTTP-заголовки запросов:
  - не встраивать напрямую на страницу или в тело электронного письма данные, полученные из недоверенных источников;
  - заменять потенциально небезопасные символы в синтаксисе HTML на их эквиваленты, которые не являются символами форматирования;
  - проверять данные как на стороне клиента, так и на стороне сервера.
- Рассмотреть возможность внедрения межсетевого экрана уровня приложений (Web Application Firewall).
- Использовать фреймворки, которые автоматически обрабатывают данные пользователей для защиты от проведения различных атак (внедрение HTML-кода, «Межсайтовое выполнение сценариев» и т. д.).

4.Перечень недостатков

4.1. Раскрытие отладочной и конфигурационной информации

Статус: ***Не исправлено***

Результат перепроверки:

Недостаток воспроизводится без изменений.

Описание:

в веб-приложении были обнаружены сценарии, раскрывающие конфигурационную и отладочную информацию. В результате обычный пользователь приложения может получить сведения о текущих настройках, окружении или отдельных компонентах, а также прочую внутреннюю информацию. Подобная информация может предоставить потенциальному злоумышленнику дополнительные данные о структуре приложения, его пользователях или используемых компонентах, что, в свою очередь, расширит область возможных атак и упростит поиск известных уязвимостей в приложении.

Риск:

- раскрытие информации о внутренней структуре и компонентах;
- раскрытие сведений о текущей конфигурации приложения.

Технические детали:

Уязвимые хосты:

- pos.gosuslugi.ru
- fkgs.gosuslugi.ru

Сценарии:

- `/og/api/v1/version`
- `/og/widgets/load-config?orgId=<ID>`
- `/lkp/fkgs/`
- `/centrifugo/connection/websocket`

Примеры эксплуатации:

Пример раскрытия версии API:

```
https://pos.gosuslugi.ru/og/api/v1/version
```

Пример раскрытия Bearer-токена:

```
https://pos.gosuslugi.ru/og/widgets/load-config?orgId=1337
```

При создании websocket-соединения приложение возвращает информацию об используемой версии:

```
https://fkgs.gosuslugi.ru/centrifugo/connection/websocket
```

Пример раскрытия API-ключа сервиса «Яндекс Карты»:

```
https://pos.gosuslugi.ru/lkp/fkgs/
```

Рисунок 30. Раскрытие версии API

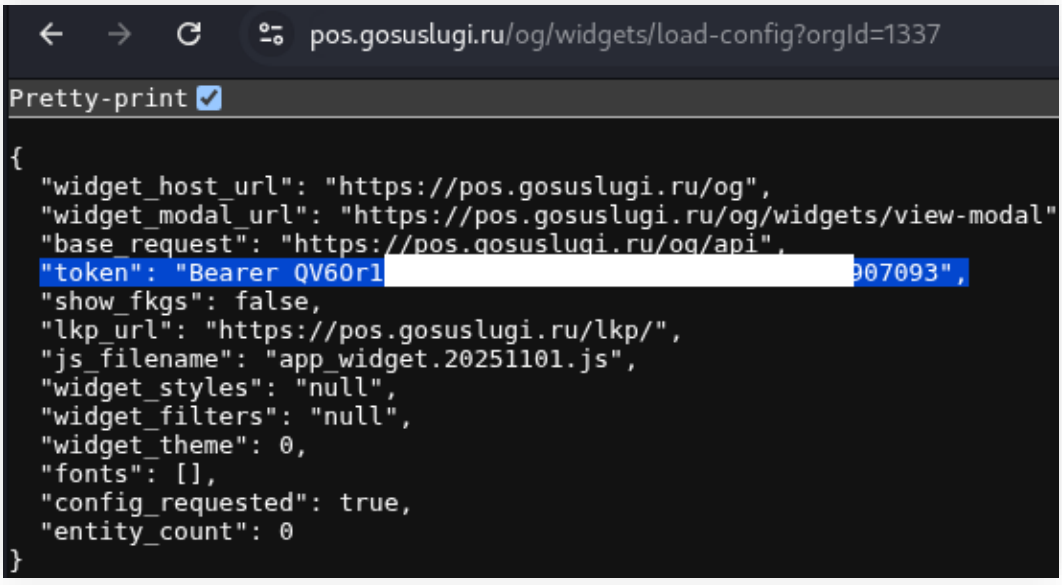

Рисунок 31. Пример раскрытия токена

Рисунок 32. Пример раскрытия версии websocket

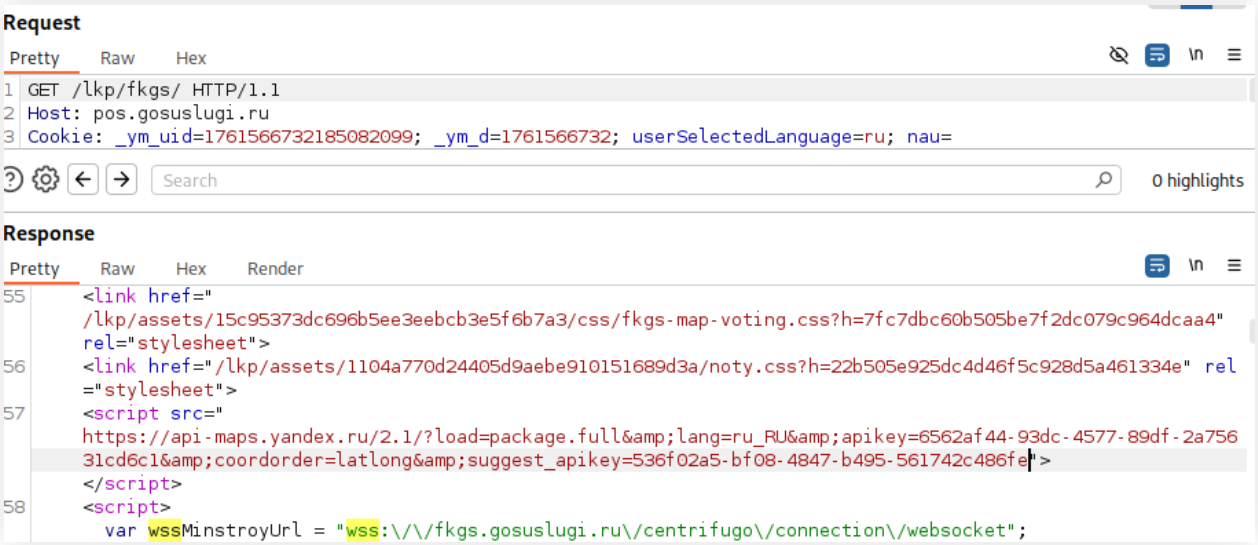

Рисунок 33. Пример раскрытия API-ключа сервиса «Яндекс Карты»

Рекомендации:

- Не раскрывать информацию о приложении, которая может быть использована при проведении атак.
- Убедиться, что функциональность отладки и диагностики отключена для приложения в продуктивном использовании.
- Ограничить доступ к файлам и сценариям, не предназначенным для пользователей приложений (документация к программным интерфейсам, журналы событий и т. д.).

4.2. Некорректная обработка ошибок

Статус: ***Исправлено***

Описание:

при возникновении исключительных ситуаций приложение возвращает сообщение об ошибке, которое содержит информацию о пользователях, компонентах или структуре приложения. При этом сообщение об ошибке может быть создано в исходном коде самого приложения (обрабатываемые разработчиками исключения) или одним из его компонентов (необработанные ошибки интерпретатора, веб-сервера, базы данных и т. д.). Подробные сообщения об ошибках могут предоставить потенциальному злоумышленнику информацию о компонентах и их версиях, абсолютные пути или прочую информацию, что упростит поиск известных уязвимостей и подготовку последующих атак.

Риск:

- раскрытие информации о внутренней структуре и компонентах;
- раскрытие сведений о текущей конфигурации приложения.

Технические детали:

Уязвимые хосты:

- pos.gosuslugi.ru

При обработках сценариев возникает ошибка, раскрывающая полный путь приложения или имена модулей приложения.

Сценарии:

- `/lkp/rating/30/`
- `/lkp/rating/38/`
- `/lkp/rating/29/`
- `/lkp/api/v1/poll/choice-dict-search/514926/`

Примеры эксплуатации:

Примеры запросов, раскрывающих полные пути приложения:

```
https://pos.gosuslugi.ru/lkp/rating/30/
https://pos.gosuslugi.ru/lkp/rating/38/
https://pos.gosuslugi.ru/lkp/rating/29/
```

Примеры запроса, раскрывающего имя модуля:

```
https://pos.gosuslugi.ru/lkp/api/v1/poll/choice-dict-search/514926/
```

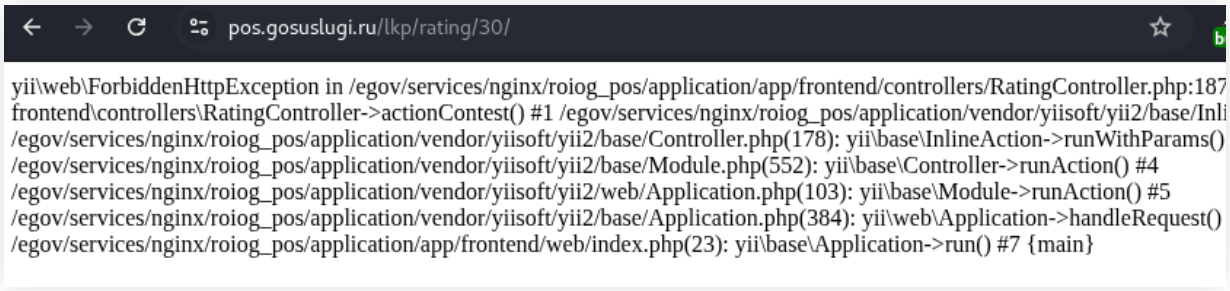

Рисунок 34. Пример раскрытия полного пути приложения

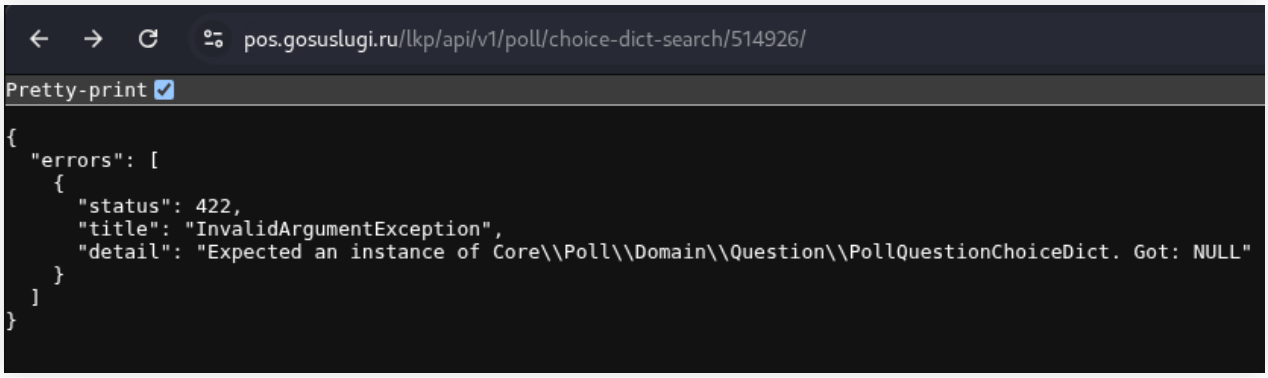

Рисунок 35. Пример сообщения об ошибке с раскрытием имени модуля

Рекомендации:

- Создать обработчик ошибок, который записывает сообщение в специальный журнал, а пользователю возвращает шаблонную страницу с кодом HTTP-ответа 200.
- Ограничить доступ к журналу ошибок для пользователей приложения.
- Возвращать в сообщениях об ошибках минимальное количество информации. Исключить вывод:
  - данных о структуре файловой системы, в том числе пути до файлов и директорий;
  - фрагментов исходного кода и конфигураций;
  - ошибок передачи запросов компонентам приложения;
  - запросов к базе данных;
  - прочей чувствительной информации.

4.3. Некорректные HTTP-заголовки безопасности

Статус: ***Не исправлено***

Результат перепроверки:

Недостаток воспроизводится без изменений.

Описание:

современные браузеры поддерживают использование множества HTTP-заголовков, часть из которых позволяет повысить безопасность веб-приложения и защитить от различных типов атак. В таблице ниже представлены некорректно настроенные HTTP-заголовки безопасности и их краткое описание. Более подробное описание и примеры использования HTTP-заголовков безопасности доступны по адресу: https://owasp.org/www-project-secure-headers/.

В таблице ниже (см. Таблица 7) приведены некорректно настроенные HTTP-заголовки безопасности.

Таблица 7. Некорректно настроенные HTTP-заголовки безопасности

HTTP-заголовок	Описание
Content-Security-Policy	Заголовок Content-Security-Policy определяет список разрешенных источников для получения ресурсов, в том числе JavaScript-сценариев. Указанный заголовок повышает устойчивость веб-приложения к атаке «Межсайтовое выполнение сценариев», поскольку запрещает выполнение сценариев, полученных из недоверенных источников

Риск:

- компрометация чувствительных данных, передаваемых между пользователем и сервером;
- выполнение действий от имени пользователей;
- проведение атак на пользователей веб-приложения;
- раскрытие чувствительных данных сторонним системам.

Технические детали:

Уязвимые хосты:

- pos.gosuslugi.ru

В приложении обнаружены некорректные настройки заголовков безопасности **Content-Security-Policy**. Директивы **unsafe-inline** и **unsafe-eval** отключают защиту **Content-Security-Policy**:

- **unsafe-inline** — разрешает выполнение встроенных JavaScript скриптов;
- **unsafe-eval** — разрешает использование eval(), setTimeout(), new Function().

Также в доверенных источниках используются широкие wildcard домены, такие как *.vk.com, *.yandex.ru и другие. Эти домены могут быть использованы злоумышленниками для обхода политик **Content-Security-Policy**.

Сценарии:

- `/lkp/fkgs/`
- `/lkp/*`

Пример эксплуатации:

```
GET /lkp/fkgs/ HTTP/1.1
Host: pos.gosuslugi.ru
```

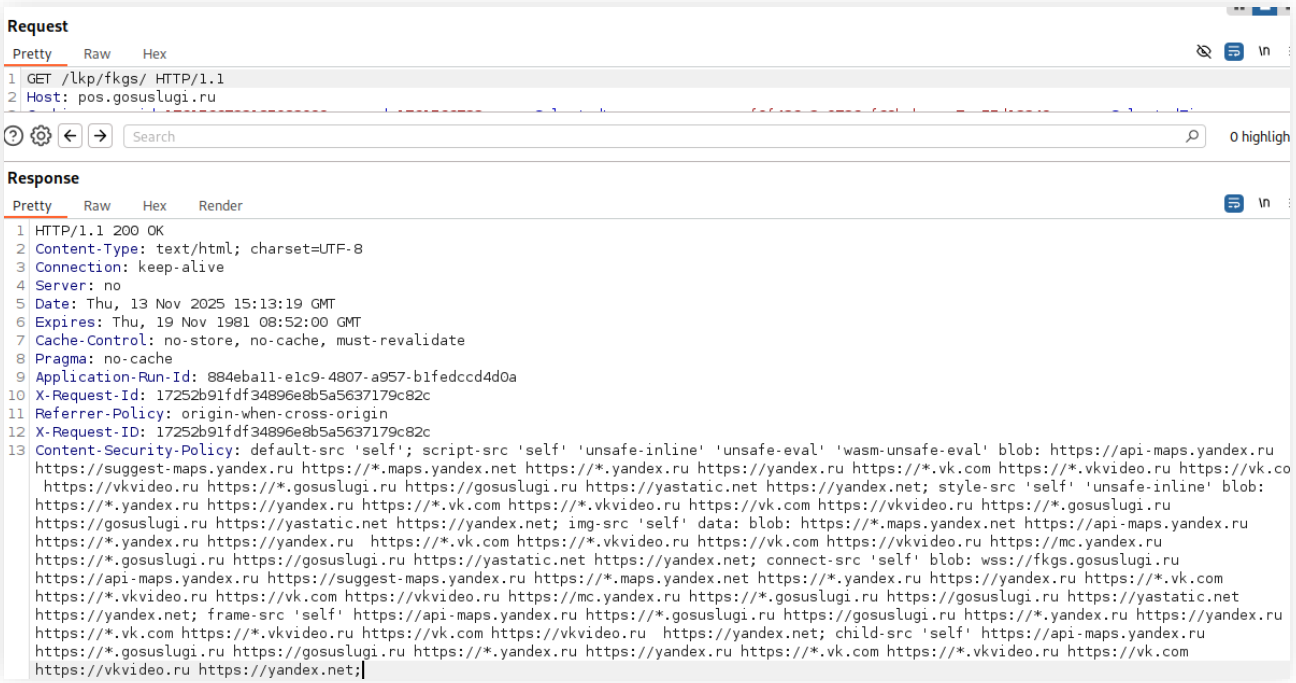

Рисунок 36. Пример используемых политик Content-Security-Policy

Рекомендации:

- Корректно настроить заголовок **Content-Security-Policy**:
  - установить директиву frame-ancestors для ограничения списка ресурсов, которые могут внедрять страницу приложения через `frame`, `iframe`, `object`, `embed` или `applet`.
  - не использовать ключевые параметры unsafe-inline и unsafe-eval. Более подробно можно ознакомиться: https://content-security-policy.com/ https://owasp.org/www-community/controls/Content_Security_Policy.

4.4. Использование тестовых данных в продуктовой среде

Статус: ***Не исправлено***

Результат перепроверки:

Недостаток воспроизводится без изменений.

Описание:

в приложении данные, созданные для целей тестирования, присутствуют или используются в рабочей (продуктовой) среде.

Уязвимость возникает из-за недостаточного контроля процессов разработки и развертывания, позволяющих смешивать тестовые артефакты с производственными данными и системами.

Риск:

- раскрытие информации, содержащейся в тестовых наборах данных.

Технические детали:

Уязвимые хосты:

- pos.gosuslugi.ru

В ходе работ были обнаружены сценарии, раскрывающие тестовые данные. Информация об указанных сценариях и раскрываемых данных приведена в таблице ниже (см. Таблица 8).

Таблица 8. Обнаруженные сценарии, раскрывающие тестовые данные

Сценарии	Раскрываемые данные
/lkp/rating/<id>	В функциональности просмотра рейтенгирований присутствует возможность перечисления по id, при этом в части идентификаторов содержатся тестовые данные.
/og/widgets/view-modal /lkp/polls/<ID>/	В приложении доступен просмотр виджетов любому пользователю, в том числе неавторизованному. При анализе доступных виджетов обнаружено множество тестовых данных. Стоит также отметить, что доступ к опросам осуществляется по простым идентификаторам, что дает возможность перечисления как активных, так и завершенных опросов.

Примеры эксплуатации:

Пример раскрытия тестовых данных в продуктовой среде рейтенгирований:

```
https://pos.gosuslugi.ru/lkp/rating/1/
```

Пример получения списка виджетов:

```
https://pos.gosuslugi.ru/og/widgets/view-modal
```

Пример получения доступа к тестовому опросу:

```
https://pos.gosuslugi.ru/lkp/polls/348364/
```

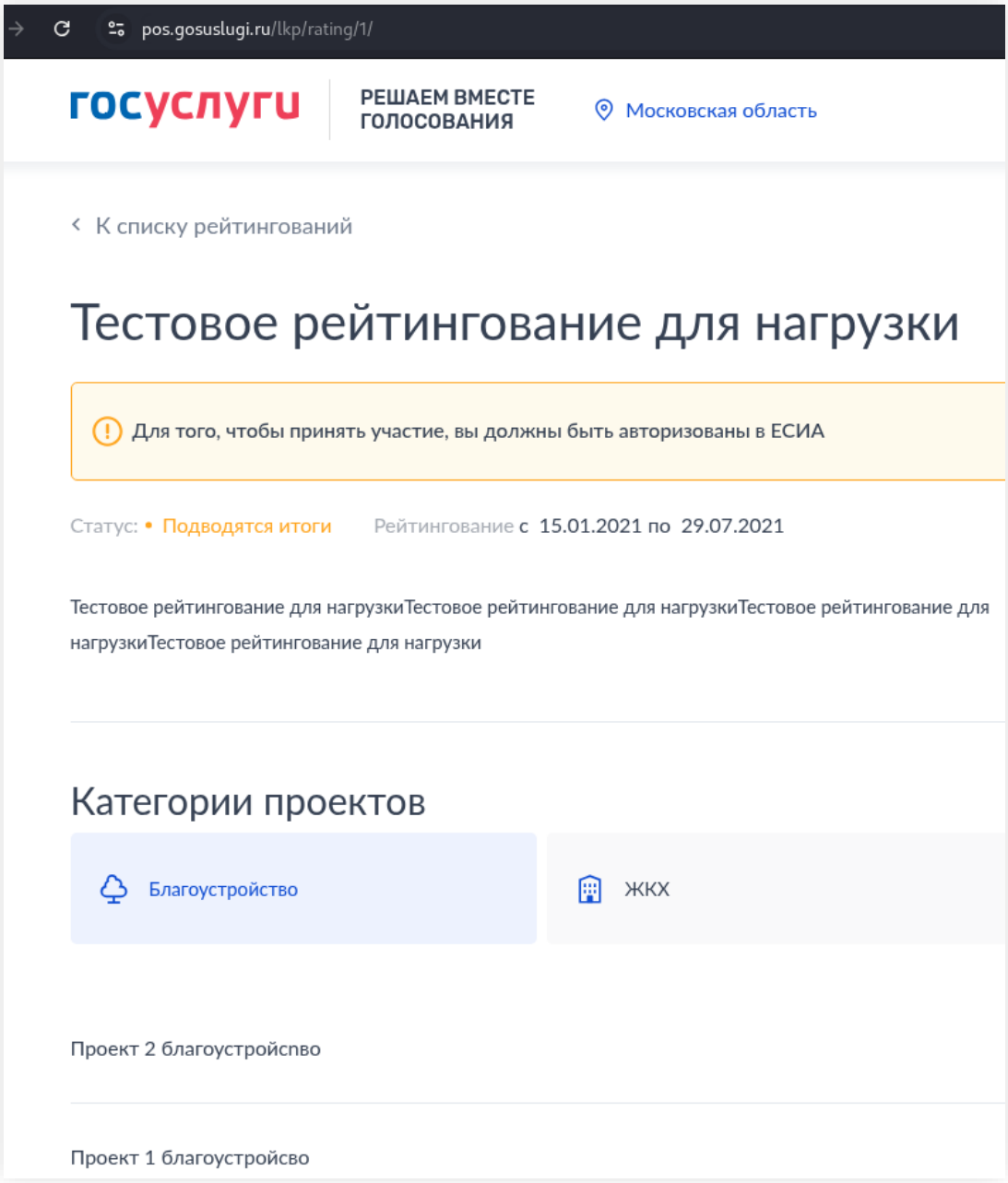

Рисунок 37. Пример тестовых данных в продуктовой среде рейтенгирований

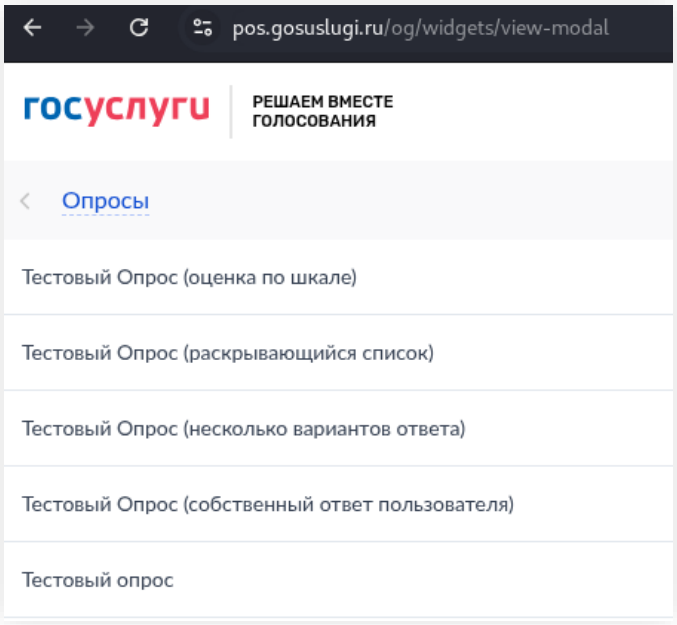

Рисунок 38. Пример получения списка виджетов

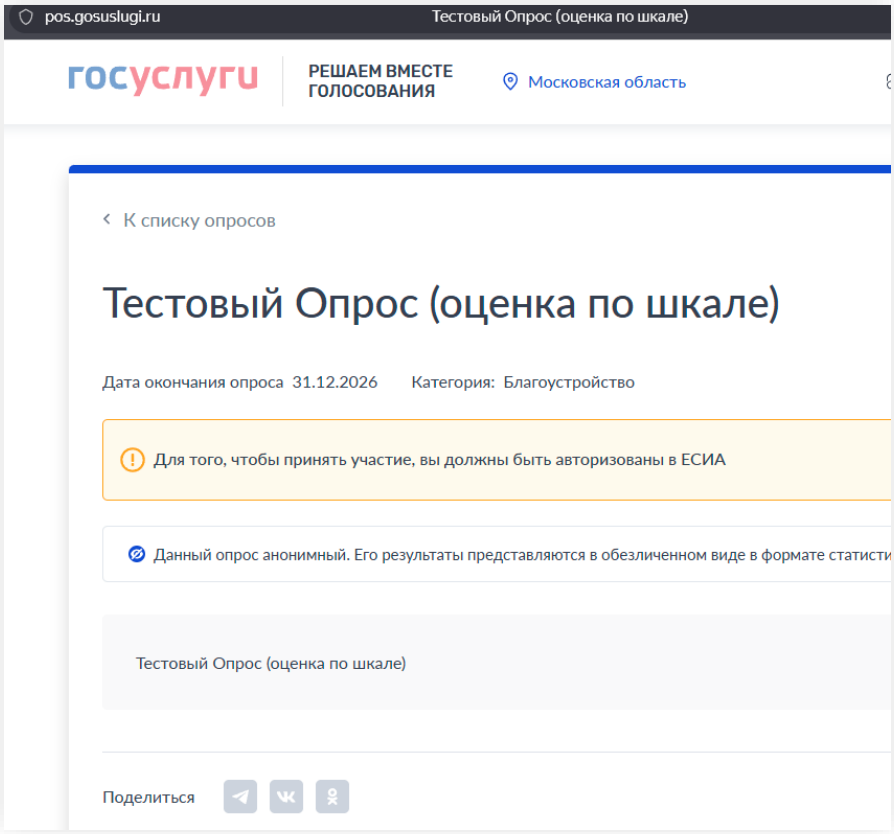

Рисунок 39. Пример получения доступа к тестовому опросу

Рекомендации:

- Удалить тестовые сценарии в продуктовой среде.

4.5. Раскрытие чувствительной информации

Статус: ***Не исправлено***

Результат перепроверки:

Недостаток воспроизводится без изменений.

Описание:

в ходе исследования было обнаружено, что в функциональности просмотра ответственных за голосование используются личные почтовые адреса.

Риск:

- раскрытие данных пользователей.

Технические детали:

Уязвимые хосты:

- pos.gosuslugi.ru

В сценариях раскрываются личные данные депутатов, а именно личные почтовые адреса.

Также с помощью перечисления идентификаторов id можно получить информацию об ответственных за территорию по короткому идентификатору, в которой будет содержаться информация о почтовом адресе.

Сценарии:

- `/lkp/fkgs/<ID>`
- `/og/improvement-minstroy/view`

Параметр: `id`

Любой пользователь приложения может получить информацию о почтовых адресах ответственных:

```
https://pos.gosuslugi.ru/lkp/fkgs/15105/96855/
```

Авторизованный пользователь приложения с правами доступа к функциональности **/og** может получить информацию о почтовых адресах ответственных по короткому идентификатору:

```
https://pos.gosuslugi.ru/og/improvement-minstroy/view?id=9211
```

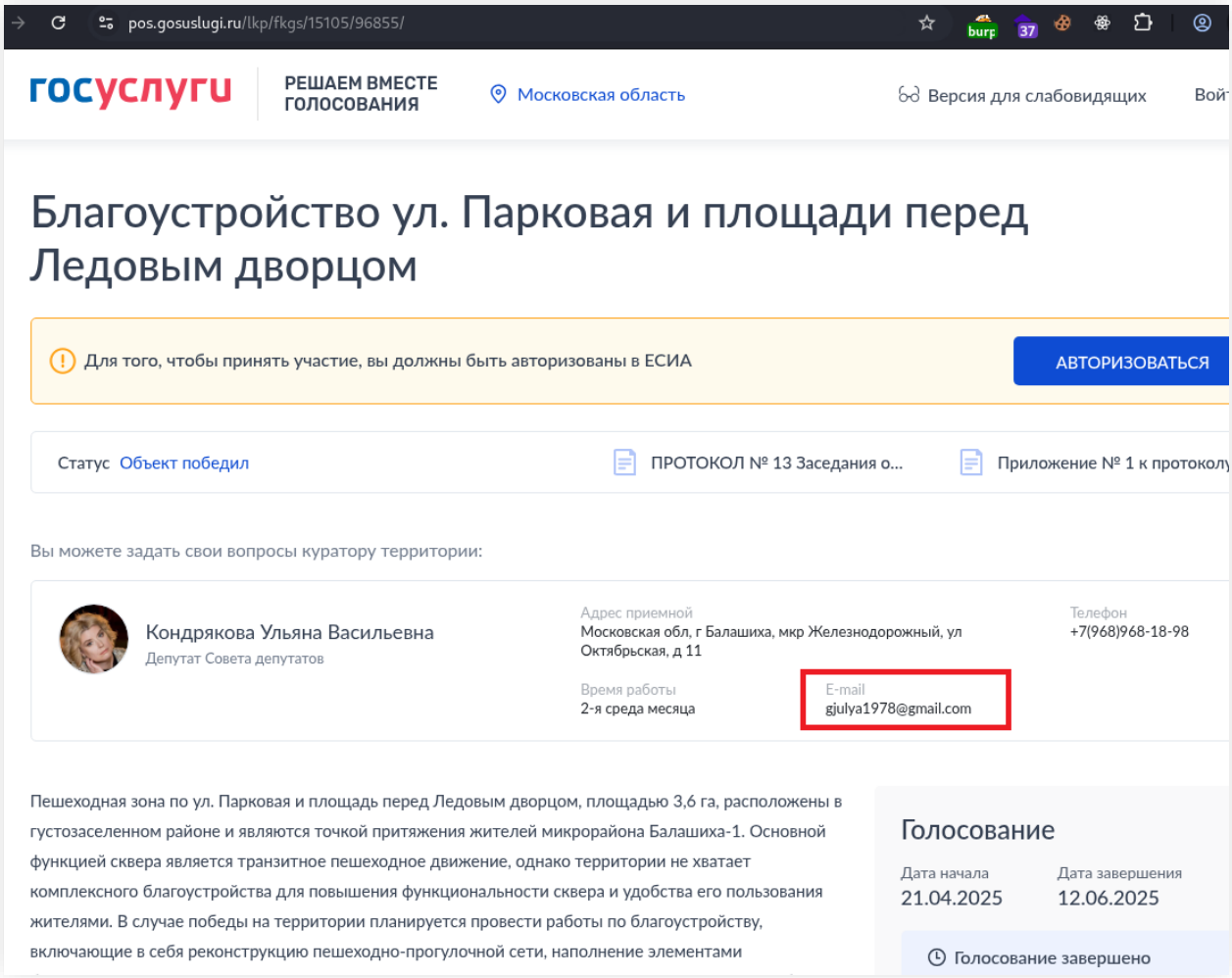

Рисунок 40. Пример использования личной почты для обратной связи (1)

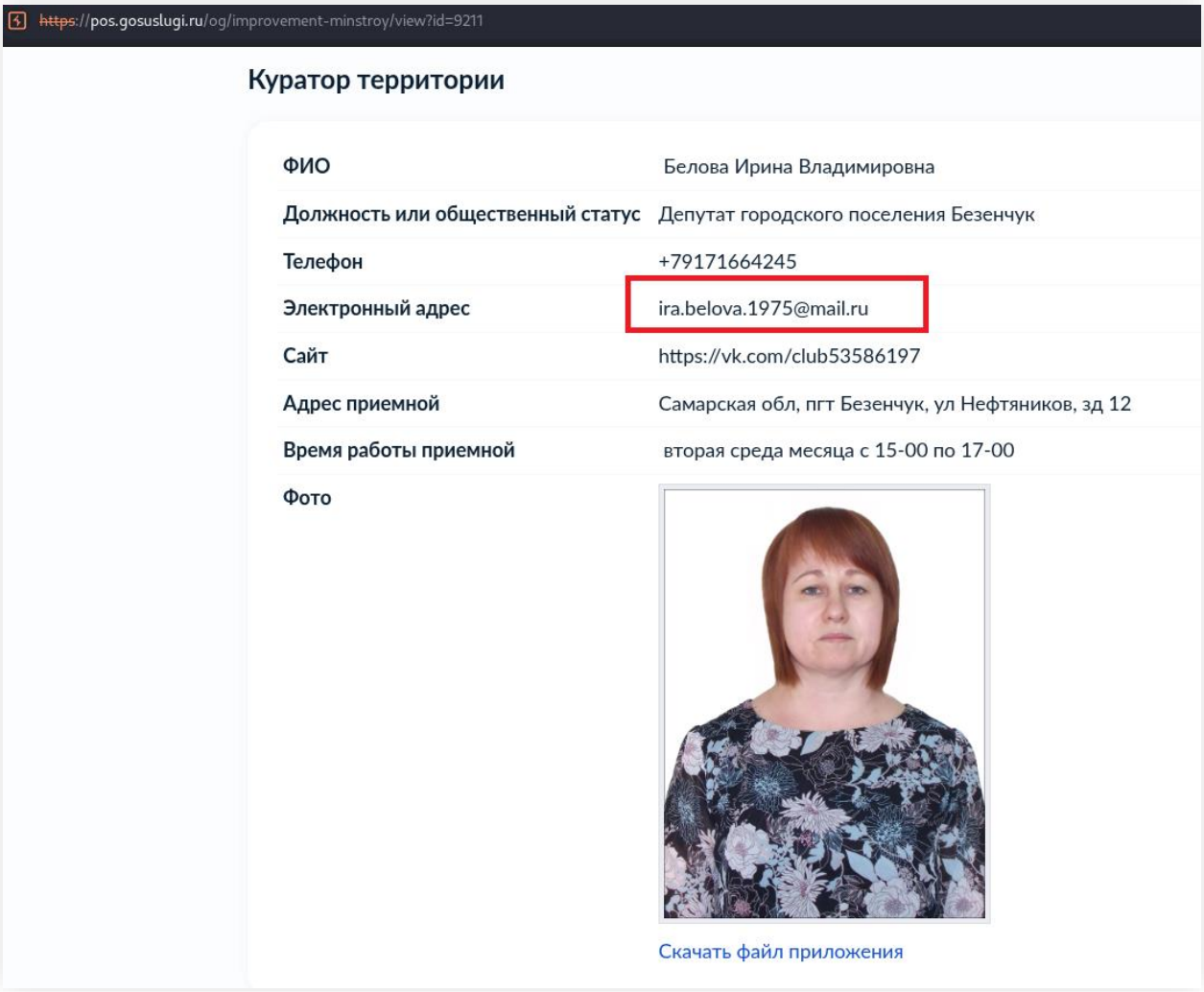

Рисунок 41. Пример использования личной почты для обратной связи (2)

Рекомендации:

- Не использовать личные почтовые адреса для связи с ответственными.

Приложение 1. Методика оценки защищенности

Методика оценка критичности эксплуатации уязвимости

Свойство «критичность» эксплуатации уязвимости характеризует последствия ее эксплуатации с точки зрения угроз нарушения конфиденциальности, целостности и доступности информации, обрабатываемой на уязвимом ресурсе. Значения и описание уровней критичности эксплуатации уязвимостей приведено в Таблице А.1.

Таблица А.1. Уровни критичности уязвимостей

Значение	Нарушение конфиденциальности	Нарушение целостности	Нарушение доступности
Информационного характера	Не происходит	Не происходит	Не происходит
Низкий	Получение нарушителем доступа к информации низкой степени критичности в результате эскалации привилегий	Нарушение целостности информации низкой степени критичности с правами обычного пользователя	Кратковременный отказ в обслуживании приложения
Средний	Нарушение конфиденциальности критичной информации с правами обычного пользователя	Нарушение целостности критичной информации с правами обычного пользователя	Длительный отказ в обслуживании критичного приложения или кратковременный отказ в обслуживании ОС
Высокий	Нарушение конфиденциальности критической информации с правами администратора	Нарушение целостности критической информации с правами администратора	Длительный отказ в обслуживании ОС

Методика оценки простоты эксплуатации уязвимости

Свойство «простота» эксплуатации уязвимости определяет, какие аппаратные и программные средства, профессиональные навыки, а также какое количество временных и вычислительных ресурсов, необходимо потенциальному нарушителю для проведения успешной эксплуатации (Таблица А.2).

Таблица А.2. Уровни простоты эксплуатации уязвимости

Значение	Описание
Низкий	Для эксплуатации уязвимости требуется разработка новых программных средств, проведение анализа конфигурации атакуемых информационных ресурсов, выявление и проверка различных возможных путей и условий успешной эксплуатации уязвимости, вычислительные мощности, временной резерв или взаимодействие с пользователем. Атакующий должен обладать значительными профессиональными навыками и знаниями в специфичных областях.

Значение	Описание
Средний	Для эксплуатации уязвимости требуется наличие специальных программных или аппаратных средств, проведение анализа конфигурации атакуемых информационных ресурсов, вычислительные мощности, временной резерв или взаимодействие с пользователем. Атакующему достаточно обладать незначительным объемом профессиональных навыков и знаний для реализации атаки.
Высокий	Для эксплуатации уязвимости не требуется использование специальных аппаратных или программных средств, значительные вычислительные мощности или временной резерв, детальное знание конфигурации атакуемых информационных ресурсов. Атакующему для реализации атаки не требуются специфичные профессиональные навыки и знания.

Методика оценки доступности

Свойство «доступность» определяет, каким классам пользователей доступен уязвимый ресурс и/или функциональность, в которой реализуется выявленная уязвимость (Таблица А.3).

Таблица А.3. Уровни доступности

Значение	Описание
Низкий	Привилегированные пользователи
Средний	Зарегистрированные пользователи
Высокий	Все пользователи

Методика оценки вероятности эксплуатации уязвимости

Вероятность эксплуатации уязвимости рассчитывается на основе простоты эксплуатации и области доступности уязвимой функциональности по таблице А.4.

Таблица А.4. Уровень вероятности эксплуатации

Вероятность эксплуатации		Простота эксплуатации		
		Низкий	**Средний**	**Высокий**
Доступность	**Низкий**	**Низкий**	**Низкий**	**Средний**
	Средний	**Низкий**	**Средний**	**Высокий**
	Высокий	**Средний**	**Высокий**	**Высокий**

Методика оценки итогового риска уязвимости

Риск уязвимости (по одной из угроз) рассчитывается на основе критичности уязвимости (по одной из угроз) и вероятности эксплуатации уязвимости по таблице А.5.

Таблица А.5. Уровень риска

Риск уязвимости	Вероятность эксплуатации

		Низкий	Средний	Высокий
Критичность уязвимости	Низкий	Низкий	Низкий	Средний
	Средний	Низкий	Средний	Высокий
	Высокий	Средний	Высокий	Высокий

Методика моделирования нарушителя

Под понятием «нарушитель» или «злоумышленник» подразумевается лицо или группа лиц, состоящих или не состоящих в сговоре, которые в результате предумышленных или непредумышленных действий потенциально могут нанести ущерб информационным ресурсам Заказчика.

Предполагается, что злоумышленник обладает квалификацией, сравнимой с квалификацией специалистов Исполнителя, имеет доступ к тем же опубликованным методикам выявления и эксплуатации уязвимостей информационных ресурсов и располагает теми же инструментальными средствами.

При проведении работ специалисты Исполнителя опираются на модель нарушителя информационной безопасности по методике ФСТЭК и проводят оценку потенциальных нарушителей по следующим характеристикам:

- Уровень осведомленности об исследуемом ресурсе, системе и/или инфраструктуре.
- Уровень прав и условий доступа к информационным ресурсам и инфраструктуре.

В зависимости **от уровня осведомленности** нарушители подразделяются на:

- Нарушитель, не обладающий документацией, исходным кодом или привилегированным доступом. Может иметь права обычного пользователя, регистрироваться и использовать стандартную функциональность.
- Нарушитель, обладающий знаниями об инфраструктуре. Пользователь, у которого нет привилегированного доступа, однако он обладает документацией и исходным кодом. Имеет права обычного пользователя, может регистрироваться и использовать стандартную функциональность.
- Нарушитель, обладающий привилегированным доступом ко всей инфраструктуре, в том числе к административному интерфейсу, а также обладающий документацией и исходным кодом. Может использовать всю имеющуюся функциональность.

В зависимости **от прав и условий доступа** к информационным ресурсам и сетям нарушители делятся на:

- **Внешний нарушитель.** Не имеет права доступа к информационным ресурсам и/или инфраструктуре или её отдельным компонентам и реализует атаки из-за ее границ.
- **Внутренний нарушитель.** В момент начала атаки находился внутри информационной сети организации и обладал определенными правами доступа к информационным ресурсам.