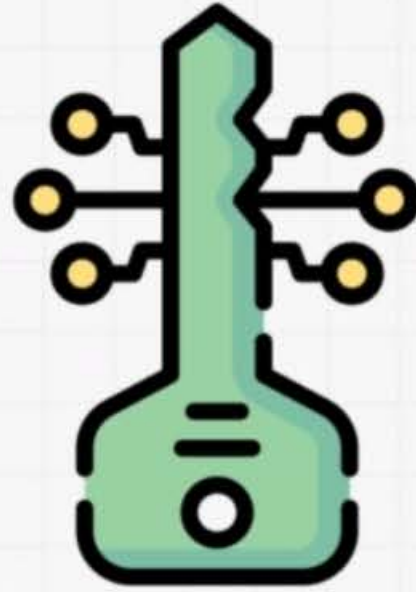


group number one, which are attacks that the user is responsible for, and group number two attacks



USER



SESSION



DEVELOPER

interested in covering mistakes in website configuration that allows us to steal our session.



SESSION FIXATION



SESSION
HIJACKING



WEAK
ENCRYPTION/WEAK
SESSION COOKIES

SESSION MANAGEMENT

Title
Authenticate-2

IP Address
10.10.54.81

Expires
58m 18s



Add 1 hour

Terminate

A lot of times what happens is that developers forgets to sanitize the input(username & password) given by the user in the code of their application, which can make them vulnerable to attacks like SQL injection. However, we are going to focus on a vulnerability that happens because of a developer's mistake but is very easy to exploit i.e re-registration of an existing user.

Let's understand this with the help of an example, say there is an existing user with the name admin and now we want to get access to their account so what we can do is try to re-register that username but with slight modification. We are going to enter " admin"(notice the space in the starting). Now when you enter that in the username field and enter other required information like email id or password and submit that data. It will actually register a new user but that user will have the same right as normal admin. That new user will also be able to see all the content presented under the user admin.

To see this in action go to <http://10.10.54.81:8888> and try to register a user name darren, you'll see that user already exists so then try to register a user " darren" and you'll see that you are now logged in and will be able to see the content present only in Darren's account which in our case is the flag that you need to retrieve.

What is the flag that you found in darren's account?

Answer format: *****

Submit

Now try to do the same trick and see if you can login as arthur.

No answer needed

Completed

What is the flag that you found in arthur's account?

Answer format: *****

Submit



Would you like Firefox to save this login for
http://10.10.54.81:8888?

darren

••••••••

☐ Show password

Don't Save



Save

Username

Password

Sign in

Register

Register

Username:

darren

Email:

test@gmail.com

Password:

••••••••

Register



10.10.54.81:8888/logged

[Kali Linux](#) [Kali Training](#) [Kali Tools](#) [Kali Forums](#) [Kali Docs](#) [NetHunter](#) [Offensive Security](#) [MSFU](#) [Exploit-DB](#) [GHDB](#)

Authentication

fe86079416a21a3c99937fea8874b667



| | |
|---------------|---|
| OWASP 2013 | A1 - Injection (SQL) |
| OWASP 2010 | A1 - Injection (Other) |
| OWASP 2007 | A2 - Broken Authentication and Session Management |
| Web Services | A3 - Cross Site Scripting (XSS) |
| HTML 5 | A4 - Insecure Direct Object References |
| Others | A5 - Security Misconfiguration |
| Documentation | A6 - Sensitive Data Exposure |
| Resources | A7 - Missing Function Level Access Control |
| | A8 - Cross Site Request Forgery (CSRF) |
| | A9 - Using Components with Known Vulnerabilities |
| | A10 - Unvalidated Redirects and Forwards |

Getting Started Project Whitepaper



Mutillidae: Deliberately Vulnerable Web Penetration Test Framework

| | |
|-----------------------|-------------------|
| Authentication Bypass | Via Brute Force |
| Privilege Escalation | Via Cookies |
| Username Enumeration | Via SQL Injection |



[Video Tutorials](#)



[Listing of vulnerabilities](#)



[Bug Report Email Address](#)



[Release Announcements](#)



[Feature Requests](#)

92.168.1.11/mutillidae/index.php?page=privilege-escalation.php

[Mutillidae Console](#)

 Request to http://192.168.1.11:80

Forward

Drop

Intercept is on

Action

Open Browser

Pretty Raw In Actions ▾

```
1 GET /mutillidae/index.php?page=privilege-escalation.php HTTP/1.1
2 Host: 192.168.1.11
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
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 Referer: http://192.168.1.11/mutillidae/index.php?popUpNotificationCode=AU1
8 Connection: close
9 Cookie: showhints=1; username=test1; uid=25; PHPSESSID=de3ta4ecu7o8ab7a2b8lj4bjh2; acopendivids=swingset,jotto,phpbb2,redmine; acgroupswithpersist=nada
10 Upgrade-Insecure-Requests: 1
11 Cache-Control: max-age=0
12
13
```


GET /mutillidae/index.php?page=privilege-escalation.php HTTP/1.1

Host: 192.168.1.11

User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8

Accept-Language: en-US,en;q=0.5

Accept-Encoding: gzip, deflate

Referer: http://192.168.1.11/mutillidae/index.php?popUpNotificationCode=AU1

Connection: close

Cookie: showhints=1; username=test1; uid=1; PHPSESSID=de3ta4ecu7o8ab7a2b8lj4bjh2; acopendivids=swingset,jotto,phpbb2,redmine; acgroupswithpersist=nada

Upgrade-Insecure-Requests: 1

Cache-Control: max-age=0



e II: Web Pwn in Mass Production

ts: Enabled (1 - 5cr1pt K1dd1e) **Lo**gged In Admin: **admin** (g0t r00t?)

s | Toggle Security | Enforce SSL | Reset DB | View Log | View Captured Data

Privilege Escalation



OWASP WebGoat v5.4

[◀ Hints ▶](#) [Show Params](#) [Show Cookies](#) [Lesson Plan](#) [Show Java](#) [Solution](#)

Basic Authentication

Introduction
General
Access Control Flaws
AJAX Security
Authentication Flaws

[Password Strength](#)[Forgot Password](#)[Basic Authentication](#)[Multi Level Login 2](#)[Multi Level Login 1](#)

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

Solution Videos

[Restart this Lesson](#)

Basic Authentication is used to protect server side resources. The web server will send a 401 authentication request with the response for the requested resource. The client side browser will then prompt the user for a user name and password using a browser supplied dialog box. The browser will base64 encode the user name and password and send those credentials back to the web server. The web server will then validate the credentials and return the requested resource if the credentials are correct. These credentials are automatically resent for each page protected with this mechanism without requiring the user to enter their credentials again.

General Goal(s):

For this lesson, your goal is to understand Basic Authentication and answer the questions below.

What is the name of the authentication header:

What is the decoded value of the authentication header:

Pretty Raw In Actions ▾

```
1 GET /WebGoat/attack?Screen=35&menu=500 HTTP/1.1
2 Host: 192.168.1.11
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
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 Referer: http://192.168.1.11/WebGoat/attack?Screen=32&menu=5
8 Authorization: Basic Z3Vlc3Q6Z3Vlc3Q=
9 Connection: close
10 Cookie: PHPSESSID=de3ta4ecu7o8ab7a2b8lj4bjh2; acopendivids=swingset,jotto,phpbb2,redmine; acgroupswithpersist=nada; JSESSIONID=1D7200C41A1FF15
11 Upgrade-Insecure-Requests: 1
12 Cache-Control: max-age=0
13
14
```

Z3Vlc3Q6Z3Vlc3Q=



guest:guest

I

Solution Videos

[Restart this Lesson](#)

STAGE 1: You are Hacker Joe and you want to steal the session from Jane. Send a prepared email to the victim which looks like an official email from the bank. A template message is prepared below, you will need to add a Session ID (SID) in the link inside the email. Alter the link to include a SID.

You are: Hacker Joe

Mail To: jane.plane@owasp.org

Mail From: admin@webgoatfinancial.com

Title:

```
<b>Dear MS. Plane</b> <br><br>During the last week we had a few  
problems with our database. We have received many complaints  
regarding incorrect account details. Please use the following link  
to verify your account data:<br><br><center><a href=/WebGoat  
/attack?Screen=56&menu=1800&SID=555> Goat Hills Financial</a>  
</center><br><br>We are sorry for the any inconvenience and thank  
you for your cooporation.<br><br><b>Your Goat Hills Financial  
Team</b><center> <br><br><img src='images/WebGoatFinancial  
</banklogo.png'></center>
```

Introduction
General
Access Control Flaws
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

[Hijack a Session](#)

[Spoof an Authentication](#)

[Cookie](#)

[Session Fixation](#)

Web Services
Admin Functions
Challenge

Solution Videos

[Restart this Lesson](#)

STAGE 3: The bank has asked you to verify your data. Log in to see if your details are correct. Your user name is **Jane** and your password is **tarzan**.

You are: Victim Jane

*** You completed stage 2!**



The image shows a web browser window displaying the 'Goat Hills Financial Human Resources' login page. The page has a header with a goat logo and the title 'Goat Hills Financial Human Resources'. Below the header is a navigation bar with a home icon and a dropdown menu. The main content area features a 'Please Login' form with two input fields: 'Enter your name:' and 'Enter your password:'. A 'Login' button is positioned below the password field. The browser's address bar shows a URL starting with 'http://localhost:8080/'.

[Introduction](#)
[General](#)
[Access Control Flaws](#)
[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](#)

[Hijack a Session](#)

[Spoof an Authentication](#)

[Cookie](#)

[Session Fixation](#)

[Web Services](#)
[Admin Functions](#)
[Challenge](#)

Solution Videos

[Restart this Lesson](#)

STAGE 4: It is time to steal the session now. Use following link to reach Goat Hills Financial.

You are: Hacker Joe



Goat Hills Financial

Human Resources



Please Login

Enter your name:

Enter your password:

Login

DashboardTargetProxyIntruderRepeaterSequencerDecoderComparerExtenderProject optionsUser options

InterceptHTTP historyWebSockets historyOptions

Request to http://192.168.1.11:80

ForwardDropIntercept is onActionOpen Browser

PrettyRawInActions

1 POST /WebGoat/attack?Screen=56&menu=1800&SID=555 HTTP/1.1

2 Host: 192.168.1.11

3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0

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: 34

9 Origin: http://192.168.1.11

10 Authorization: Basic Z3Vlc3Q6Z3Vlc3Q=

11 Connection: close

12 Referer: http://192.168.1.11/WebGoat/attack?Screen=56&menu=1800&SID=NOVALIDSESSION

13 Cookie: JSESSIONID=8B92A7A7DA6AEF314BBB5F5185AF2BB6; acopendivids=swingset,jotto,phpbb2,redmine; acgroupswithpersist=nada

14 Upgrade-Insecure-Requests: 1

15

16 user3=test&pass3=test&Submit=Login

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

[Hijack a Session](#)

[Spoof an Authentication](#)

[Cookie](#)



[Session Fixation](#)

Web Services
Admin Functions
Challenge

STAGE 4: It is time to steal the session now. Use following link to reach Goat Hills Financial.

You are: Hacker Joe

*** Congratulations. You have successfully completed this lesson.**



Goat Hills Financial

Human Resources



| | |
|----------------------------|----------|
| Firstname: | Jane |
| Lastname: | Plane |
| Credit Card Type: | MC |
| Credit Card Number: | 74589864 |

Logout