

Access Control and SSRF - Project Pentesting

Access Control and SSRF Project Pentesting

Amanuel Haile

December 3rd, 2024

VULNERABILITY ASSESSMENT AND SYSTEMS ASSURANCE REPORT

TABLE OF CONTENTS

<u>Section</u>	<u>Page #</u>
1.0 General Information	3
1.1 Purpose	3
2.0 SSRF	4
2.1 Vulnerability	4
2.2 Fix	7
3.0 IDOR Information Disclosure	9
3.1 Vulnerability	9
3.2 Fix	10
4.0 IDOR Access Control	11
4.1 Vulnerability	11
4.2 Fix	14
5.0 Apache Shiro Access Control File	16
5.1 Before Fix	16
5.2 After Fix	17

1.0 General Information

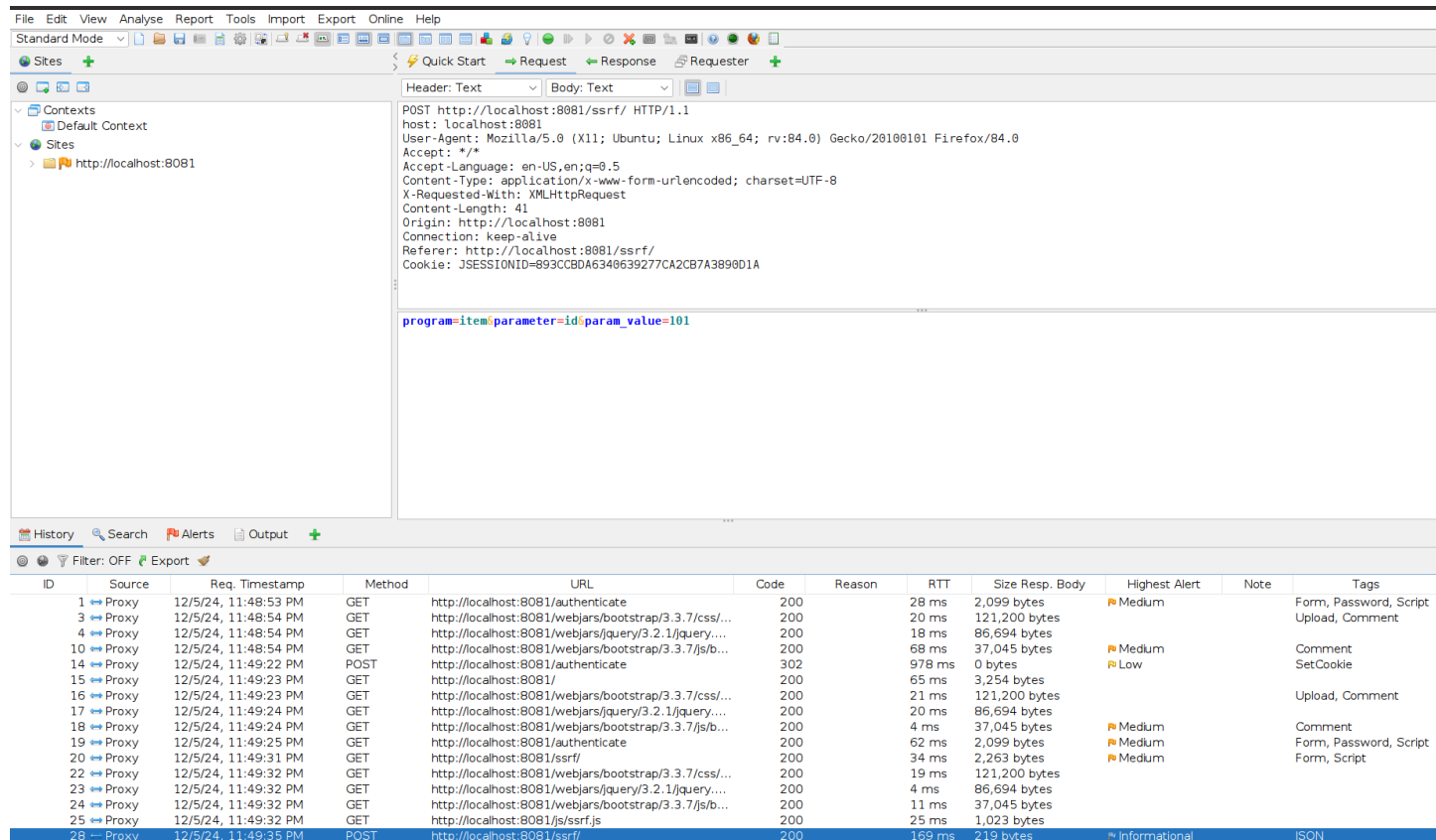
1.1 Purpose

The purpose of this vulnerability assessment and penetration test is to analyze the security of this application. The objective of this report is to discover and demonstrate the exploitation of various security vulnerabilities, specifically SSRF, Insecure Direct Object Reference (Information Disclosure and Access Control vulnerabilities), and improper implementation of security controls.

2.0 SSRF

2.1 SSRF - Vulnerability

Original POST request body:

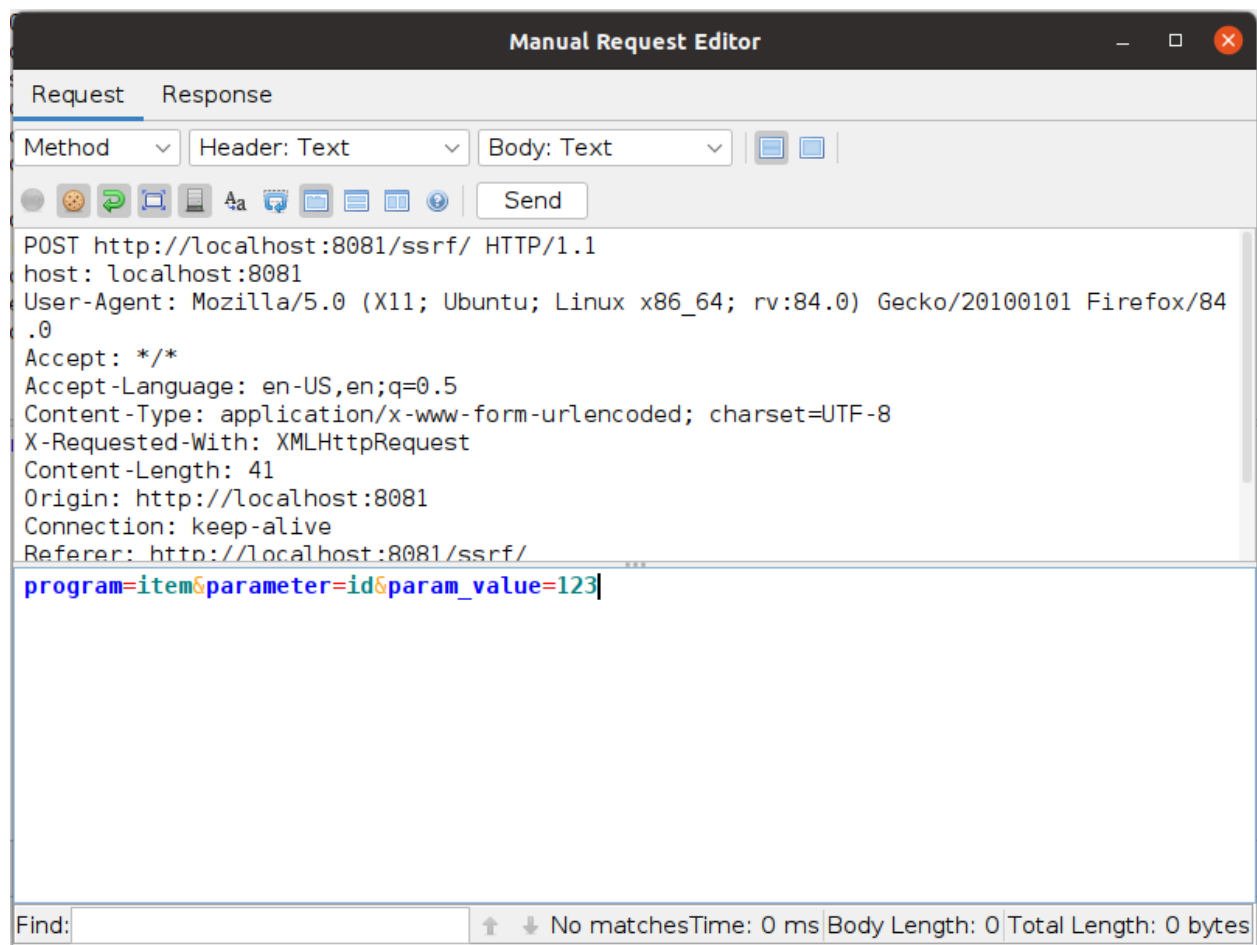


ID	Source	Req. Timestamp	Method	URL	Code	Reason	RTT	Size Resp. Body	Highest Alert	Note	Tags
1	Proxy	12/5/24, 11:48:53 PM	GET	http://localhost:8081/authenticate	200		28 ms	2,099 bytes	Medium		Form, Password, Script
3	Proxy	12/5/24, 11:48:54 PM	GET	http://localhost:8081/webjars/bootstrap/3.3.7/css/...	200		20 ms	121,200 bytes			Upload, Comment
4	Proxy	12/5/24, 11:48:54 PM	GET	http://localhost:8081/webjars/jquery/3.2.1/jquery....	200		18 ms	86,694 bytes			
10	Proxy	12/5/24, 11:48:54 PM	GET	http://localhost:8081/webjars/bootstrap/3.3.7/js/b...	200		68 ms	37,045 bytes	Medium		Comment
14	Proxy	12/5/24, 11:49:22 PM	POST	http://localhost:8081/authenticate	302		978 ms	0 bytes	Low		SetCookie
15	Proxy	12/5/24, 11:49:23 PM	GET	http://localhost:8081/	200		65 ms	3,254 bytes			
16	Proxy	12/5/24, 11:49:23 PM	GET	http://localhost:8081/webjars/bootstrap/3.3.7/css/...	200		21 ms	121,200 bytes			Upload, Comment
17	Proxy	12/5/24, 11:49:24 PM	GET	http://localhost:8081/webjars/jquery/3.2.1/jquery....	200		20 ms	86,694 bytes			
18	Proxy	12/5/24, 11:49:24 PM	GET	http://localhost:8081/webjars/bootstrap/3.3.7/js/b...	200		4 ms	37,045 bytes	Medium		Comment
19	Proxy	12/5/24, 11:49:25 PM	GET	http://localhost:8081/authenticate	200		62 ms	2,099 bytes	Medium		Form, Password, Script
20	Proxy	12/5/24, 11:49:31 PM	GET	http://localhost:8081/ssrf/	200		34 ms	2,263 bytes	Medium		Form, Script
22	Proxy	12/5/24, 11:49:32 PM	GET	http://localhost:8081/webjars/bootstrap/3.3.7/css/...	200		19 ms	121,200 bytes			
23	Proxy	12/5/24, 11:49:32 PM	GET	http://localhost:8081/webjars/jquery/3.2.1/jquery....	200		4 ms	86,694 bytes			
24	Proxy	12/5/24, 11:49:32 PM	GET	http://localhost:8081/webjars/bootstrap/3.3.7/js/b...	200		11 ms	37,045 bytes			
25	Proxy	12/5/24, 11:49:32 PM	GET	http://localhost:8081/js/ssrf.js	200		25 ms	1,023 bytes			
28	Proxy	12/5/24, 11:49:35 PM	POST	http://localhost:8081/ssrf/	200		169 ms	219 bytes	Informational		JSON

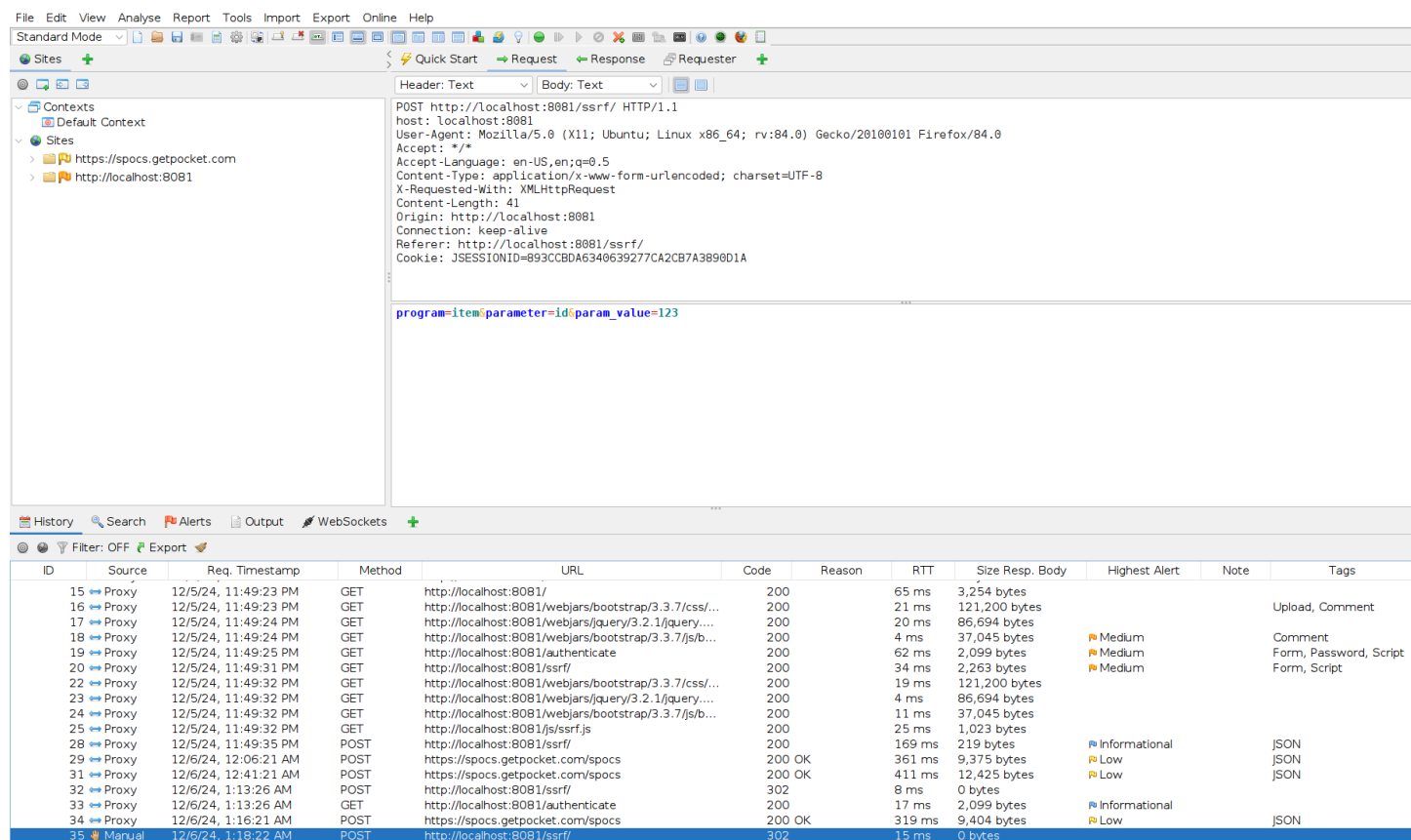
Vulnerable Code (the highlighted line):

```
public Object product_search_post(@RequestParam String program, @RequestParam String parameter, @RequestParam String param_value) {
    Map<String, String> response_data = new HashMap<String, String>();
    String msg = " ";
    try {
        URL obj = new URL("http://localhost:8081/ssrf/product/" + program + "/" + "?" + parameter + "=" + param_value);
        HttpURLConnection con = (HttpURLConnection) obj.openConnection();
        con.setRequestMethod("POST");
        con.setDoOutput(true);
        OutputStream os = con.getOutputStream();
```

This code can be exploited by sending a request with changed parameters as seen below:



Here is the screenshot showing that the request was successfully sent:



The screenshot displays the Burp Suite interface. The top menu bar includes File, Edit, View, Analyse, Report, Tools, Import, Export, Online, and Help. The main window is divided into three panes. The left pane shows the 'Sites' tab with a tree view containing 'Contexts' and 'Sites'. The middle pane shows the 'Request' tab with a 'Header: Text' view. The right pane shows the 'Body: Text' view. The request is a POST to http://localhost:8081/ssrf/ with a body containing the payload: program=item¶meter=id¶m_value=123. The status bar at the bottom shows the request was successful (200 OK).

Header: Text

Body: Text

POST http://localhost:8081/ssrf/ HTTP/1.1
Host: localhost:8081
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:84.0) Gecko/20100101 Firefox/84.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Content-Length: 41
Origin: http://localhost:8081
Connection: keep-alive
Referer: http://localhost:8081/ssrf/
Cookie: JSESSIONID=893CCBDA6340639277CA2CB7A3890D1A

program=item¶meter=id¶m_value=123

ID	Source	Req. Timestamp	Method	URL	Code	Reason	RTT	Size Resp. Body	Highest Alert	Note	Tags
15	Proxy	12/5/24, 11:49:23 PM	GET	http://localhost:8081/	200		65 ms	3,254 bytes			
16	Proxy	12/5/24, 11:49:23 PM	GET	http://localhost:8081/webjars/bootstrap/3.3.7/css/...	200		21 ms	121,200 bytes			Upload, Comment
17	Proxy	12/5/24, 11:49:24 PM	GET	http://localhost:8081/webjars/jquery/3.2.1/jquery....	200		20 ms	86,694 bytes			
18	Proxy	12/5/24, 11:49:24 PM	GET	http://localhost:8081/webjars/bootstrap/3.3.7/js/b...	200		4 ms	37,045 bytes	Medium		Comment
19	Proxy	12/5/24, 11:49:25 PM	GET	http://localhost:8081/authenticate	200		62 ms	2,099 bytes	Medium		Form, Password, Script
20	Proxy	12/5/24, 11:49:31 PM	GET	http://localhost:8081/ssrf/	200		34 ms	2,263 bytes	Medium		Form, Script
22	Proxy	12/5/24, 11:49:32 PM	GET	http://localhost:8081/webjars/bootstrap/3.3.7/css/...	200		19 ms	121,200 bytes			
23	Proxy	12/5/24, 11:49:32 PM	GET	http://localhost:8081/webjars/jquery/3.2.1/jquery....	200		4 ms	86,694 bytes			
24	Proxy	12/5/24, 11:49:32 PM	GET	http://localhost:8081/webjars/bootstrap/3.3.7/js/b...	200		11 ms	37,045 bytes			
25	Proxy	12/5/24, 11:49:32 PM	GET	http://localhost:8081/js/ssrf.js	200		25 ms	1,023 bytes			
28	Proxy	12/5/24, 11:49:35 PM	POST	http://localhost:8081/ssrf/	200		169 ms	219 bytes	Informational		JSON
29	Proxy	12/6/24, 12:06:21 AM	POST	https://spocs.getpocket.com/spocs	200 OK		361 ms	9,375 bytes	Low		JSON
31	Proxy	12/6/24, 12:41:21 AM	POST	https://spocs.getpocket.com/spocs	200 OK		411 ms	12,425 bytes	Low		JSON
32	Proxy	12/6/24, 1:13:26 AM	POST	http://localhost:8081/ssrf/	302		8 ms	0 bytes			
33	Proxy	12/6/24, 1:13:26 AM	GET	http://localhost:8081/authenticate	200		17 ms	2,099 bytes	Informational		
34	Proxy	12/6/24, 1:16:21 AM	POST	https://spocs.getpocket.com/spocs	200 OK		319 ms	9,404 bytes	Low		JSON
35	Manual	12/6/24, 1:18:22 AM	POST	http://localhost:8081/ssrf/	302		15 ms	0 bytes			

2.2 SSRF - Fix

Here is the fixed code (in the highlighted area):

```
@PostMapping("/")
@ResponseBody
public Object product_search_post(@RequestParam String program, @RequestParam String parameter, @RequestParam String param_value) {
    Map<String, String> response_data = new HashMap<>();
    String msg = "";
    final List<String> allowedPrograms = Arrays.asList("item", "design");
    final List<String> allowedParameters = Arrays.asList("id", "type");

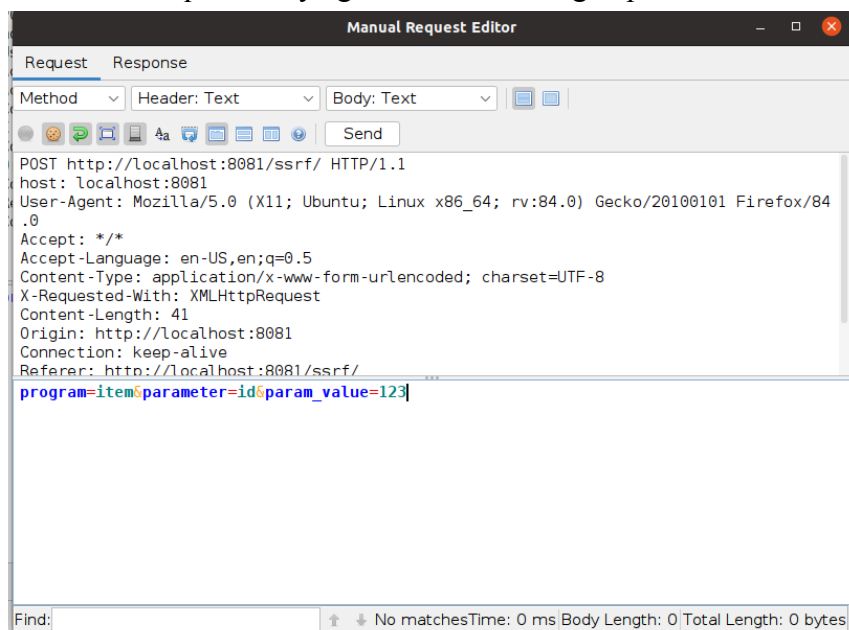
    try {
        if (!allowedPrograms.contains(program)) {
            throw new IllegalArgumentException("Invalid program value: " + program);
        }
        if (!allowedParameters.contains(parameter)) {
            throw new IllegalArgumentException("Invalid parameter value: " + parameter);
        }
        String baseUrl = "http://localhost:8081/ssrf/product/";
        String safeUrl = String.format("%s%s/?%s=%s", baseUrl, program, parameter, param_value);

        URL obj = new URL(safeUrl);
        HttpURLConnection con = (HttpURLConnection) obj.openConnection();
        con.setRequestMethod("POST");
        con.setDoOutput(true);
        try (OutputStream os = con.getOutputStream()) {
            os.flush();
        }

        int responseCode = con.getResponseCode();
    }
```

The fix involves adding validation for the program and parameter values by whitelisting only trusted options in the drop down menu. This ensures that the application only sends requests to the correct endpoints, which fixes the SSRF vulnerability.

Here is the request body again with the changed parameters:



And here is the result of sending the new request with the updated code:

The screenshot shows the Burp Suite interface. The 'Manual Request Editor' window is open, displaying the response of the POST request. The status is '200 OK'. The headers include 'HTTP/1.1 200 OK', 'X-Content-Type-Options: nosniff', 'X-XSS-Protection: 1; mode=block', 'Cache-Control: no-cache, no-store, max-age=0, must-revalidate', 'Pragma: no-cache', 'Expires: 0', 'X-Frame-Options: DENY', 'Content-Type: application/json; charset=UTF-8', 'Date: Fri, 06 Dec 2024 07:06:04 GMT', and 'Content-Length: 33'. The body is '{\"msg\": \"null\", \"status\": \"failure\"}'. A red arrow points to the 'status' field in the JSON body. The background shows the Burp Suite interface with a list of requests and responses.

ID	Source	Req. Timestamp	Method	URL	Code	Reason	RTT	Size Resp. Body	Highest Alert	Note	Tags
45	Proxy	12/6/24, 1:45:21 AM	GET	http://localhost:8081/webjars/bootstrap/3.3.7/css/...	200		31 ms	121,200 bytes			
47	Proxy	12/6/24, 1:45:21 AM	GET	http://localhost:8081/authenticate	200		37 ms	2,099 bytes	Medium		Form, Password, Script
48	Proxy	12/6/24, 1:45:24 AM	GET	http://localhost:8081/ssrf/	200		20 ms	2,263 bytes	Medium		Form, Script
49	Proxy	12/6/24, 1:45:24 AM	GET	http://localhost:8081/webjars/bootstrap/3.3.7/css/...	200		15 ms	121,200 bytes			
50	Proxy	12/6/24, 1:45:24 AM	GET	http://localhost:8081/webjars/jquery/3.2.1/jquery....	200		18 ms	86,694 bytes	Medium		Form, Script, Comment
51	Proxy	12/6/24, 1:45:24 AM	GET	http://localhost:8081/webjars/bootstrap/3.3.7/js/b...	200		13 ms	37,045 bytes	Medium		Comment
52	Proxy	12/6/24, 1:45:24 AM	GET	http://localhost:8081/js/ssrf.js	200		17 ms	1,023 bytes	Informational		
53	Proxy	12/6/24, 2:05:16 AM	POST	http://localhost:8081/ssrf/	200		117 ms	219 bytes	Informational		JSON
54	Manual	12/6/24, 1:49:24 AM	POST	http://localhost:8081/ssrf/	200		26 ms	33 bytes	Informational		JSON
55	Proxy	12/6/24, 1:51:21 AM	POST	https://spocs.getpocket.com/spocs	200 OK		1.45 s	9,394 bytes	Low		
56	Proxy	12/6/24, 1:54:26 AM	POST	http://localhost:8081/ssrf/	200		20 ms	219 bytes	Informational		JSON
57	Proxy	12/6/24, 2:05:16 AM	GET	http://localhost:8081/ssrf/	200		750 ms	2,263 bytes	Medium		Form, Script
58	Proxy	12/6/24, 2:05:16 AM	GET	http://localhost:8081/webjars/jquery/3.2.1/jquery....	200		45 ms	86,694 bytes			
59	Proxy	12/6/24, 2:05:16 AM	GET	http://localhost:8081/webjars/jquery/3.2.1/jquery....	200		48 ms	1,023 bytes			
61	Proxy	12/6/24, 2:05:16 AM	GET	http://localhost:8081/webjars/bootstrap/3.3.7/css/...	200		113 ms	121,200 bytes	Informational		
60	Proxy	12/6/24, 2:05:16 AM	GET	http://localhost:8081/webjars/bootstrap/3.3.7/js/b...	200		63 ms	37,045 bytes			
62	Proxy	12/6/24, 2:05:20 AM	POST	http://localhost:8081/ssrf/	200		122 ms	219 bytes	Informational		JSON
63	Manual	12/6/24, 2:06:04 AM	POST	http://localhost:8081/ssrf/	200		25 ms	33 bytes	Informational		JSON

3.0 IDOR Information Disclosure

3.1 IDOR Information Disclosure - Vulnerability

There is an information disclosure vulnerability in the “Get Profile” Button on this app. When a user logs into the “idor” page and goes to the “Customer Profile” section, a button with the name “Get Profile” appears. If the victim decides to click on the “Get Profile” button, this information is shown:

Get Profile: There is often data in the raw response that doesn't show up on the screen/page. Please change code accordingly to apply restrictions on viewing all data.

Get Profile

Name	Email	Phone
Moumita Das	mpurba@xyz.com	980-126-5874

The issue is that this information is shown in ZAP when the same GET request is sent again through the ZAP app, which shows that the server is sending more information than what is necessary to the user:

```
HTTP/1.1 200
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-Frame-Options: DENY
Content-Type: application/json; charset=UTF-8
Content-Length: 214
Date: Sat, 07 Dec 2024 10:06:45 GMT
```

```
[{"id": "886459", "password": "123", "name": "Moumita Das", "phone": "980-126-5874", "amount": "$52,000", "performance": "Job knowledge: F</br>Work quality: S</br>Attendance: E</br>Communication: S", "email": "mpurba@xyz.com"}]
```

3.2 IDOR Information Disclosure - Fix

Here is the vulnerable code for the Information Disclosure vulnerability:

```
List list = new ArrayList<>();
String json = "";
try {
    list.add(customer_sessionInfo);
    json = objectMapper.writeValueAsString(list);
} catch (JsonProcessingException e) {
    json = "{\"status\":\"error\"}";
    e.printStackTrace();
}
return json;
```

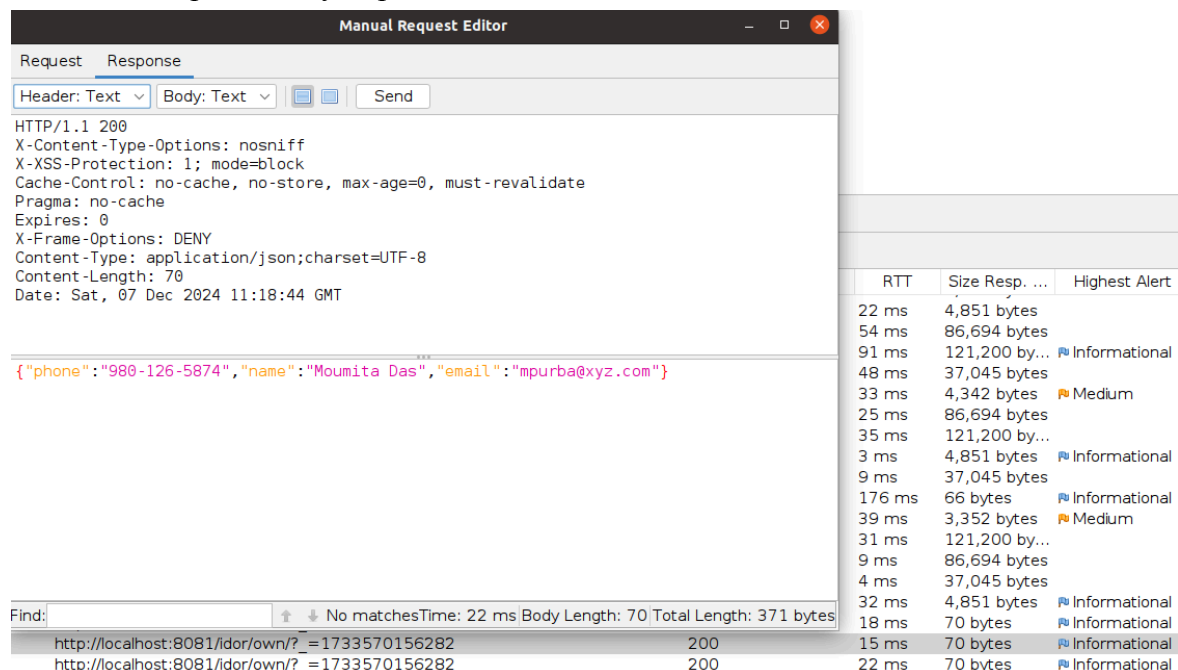
The issue is that the password is included in the JSON response, which exposes it to the client.

And here is the updated code:

```
@GetMapping("/own")
@ResponseBody
public String idor_profile_own(HttpServletRequest request) throws Exception {
    HttpSession session = request.getSession();
    ObjectMapper objectMapper = new ObjectMapper();
    if (session.getAttribute("logged_in_customer") == null) {
        Map<String, String> response_data = new HashMap<>();
        response_data.put("status", "error");
        response_data.put("msg", "Please login as customer first");
        return objectMapper.writeValueAsString(response_data);
    }
    Map<String, String> customer_sessionInfo = (Map<String, String>) session.getAttribute("logged_in_customer");
    Map<String, String> sanitizedResponse = new HashMap<>();
    sanitizedResponse.put("name", customer_sessionInfo.get("name"));
    sanitizedResponse.put("email", customer_sessionInfo.get("email"));
    sanitizedResponse.put("phone", customer_sessionInfo.get("phone"));
    return objectMapper.writeValueAsString(sanitizedResponse);
}
```

The method now only includes the name, email, and phone in the JSON response. Sensitive data such as id, password, and amount are not included.

Here is the response body to prove that this fix worked:



The screenshot shows the Manual Request Editor interface. The 'Response' tab is selected, and the 'Body: Text' field is active. The response body is a JSON object: `{"phone": "980-126-5874", "name": "Moumita Das", "email": "mpurba@xyz.com"}`. The 'Find' field is empty, and the 'No matches' message is displayed. The 'Time' field shows 22 ms, 'Body Length' is 70, and 'Total Length' is 371 bytes.

RTT	Size Resp. ...	Highest Alert
22 ms	4,851 bytes	
54 ms	86,694 bytes	
91 ms	121,200 by...	Informational
48 ms	37,045 bytes	
33 ms	4,342 bytes	Medium
25 ms	86,694 bytes	
35 ms	121,200 by...	
3 ms	4,851 bytes	Informational
9 ms	37,045 bytes	
176 ms	66 bytes	Informational
39 ms	3,352 bytes	Medium
31 ms	121,200 by...	
9 ms	86,694 bytes	
4 ms	37,045 bytes	
32 ms	4,851 bytes	Informational
18 ms	70 bytes	Informational
15 ms	70 bytes	Informational
22 ms	70 bytes	Informational

4.0 IDOR Access Control

4.1 IDOR Access Control - Vulnerability

There is also a access control vulnerability in the Performance Evaluation button, and the information that would usually be shown to the user can be seen below:

Performance Evaluation: View someone else's result by man in the middle attack. Please change code accordingly to apply restrictions on viewing other's data.

Performance Evaluation

Name	Email	Performance Evaluation(%)
Moumita Das	mpurba@xyz.com	Job knowledge: F Work quality: S Attendance: E Communication: S

On ZAP, if the customer_id is changed to a different user's, you will be able to see their information instead, which can be seen below:

Original GET request body:

Manual Request Editor

RequestResponse

MethodHeader: TextBody: TextSend

GET http://localhost:8081/idor/info/?requested_customer_id=886459&_=1733567480907 HTTP/1.1
host: localhost:8081
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:84.0) Gecko/20100101 Firefox/84.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: en-US,en;q=0.5
X-Requested-With: XMLHttpRequest
Connection: keep-alive
Referer: http://localhost:8081/idor/profile/
Cookie: JSESSIONID=11D184014746E006E63083CFADE8AD9D
content-length: 0

	RTT	Size Resp. ...	Highest Alert
	5 ms	37,045 bytes	Informational
	2 ms	4,851 bytes	Informational
	4 ms	180 bytes	Informational
	5 ms	179 bytes	Informational
	23 ms	180 bytes	Informational
	16 ms	4,342 bytes	Medium
	7 ms	121,200 by...	
	8 ms	86,694 bytes	Medium
	24 ms	37,045 bytes	
	2 ms	4,851 bytes	Informational
	11 ms	66 bytes	Informational
	16 ms	3,352 bytes	Medium
	4 ms	121,200 by...	
	23 ms	86,694 bytes	Medium
	5 ms	37,045 bytes	
	2 ms	4,851 bytes	Informational
	3 ms	179 bytes	Informational
	23 ms	179 bytes	Informational

Original Response Body:

The screenshot shows the 'Manual Request Editor' window with the 'Response' tab selected. The response headers are: HTTP/1.1 200, X-Content-Type-Options: nosniff, X-XSS-Protection: 1; mode=block, Cache-Control: no-cache, no-store, max-age=0, must-revalidate, Pragma: no-cache, Expires: 0, X-Frame-Options: DENY, Content-Type: application/json; charset=UTF-8, Content-Length: 179, Date: Sat, 07 Dec 2024 10:35:23 GMT. The response body is a JSON object: {"msg": "You have successfully seen data", "performance": {"Job knowledge: F</br>Work quality: S</br>Attendance: E</br>Communication: S", "name": "Moumita Das", "email": "mpurba@xyz.com"}}. A table on the right shows RTT values for various requests. The status bar at the bottom indicates 'No matches' and provides timing and length information for the selected request.

RTT
4 ms
5 ms
23 ms
16 ms
7 ms
8 ms
24 ms
2 ms
11 ms
16 ms
4 ms
23 ms
5 ms
2 ms
3 ms
23 ms
11 ms
11 ms

Find: No matchesTime: 11 msBody Length: 179Total Length: 481 bytes

http://localhost:8081/idor/info/?requested_customer_id=886459&_=17... 200

Changed customer_id:

The screenshot shows the 'Manual Request Editor' window with the 'Request' tab selected. The request method is GET, and the URL is http://localhost:8081/idor/info/?requested_customer_id=886359&_=1733567289836. The request headers are: host: localhost:8081, User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:84.0) Gecko/20100101 Firefox/84.0, Accept: application/json, text/javascript, */*; q=0.01, Accept-Language: en-US,en;q=0.5, X-Requested-With: XMLHttpRequest, Connection: keep-alive, Referer: http://localhost:8081/idor/profile/, Cookie: JSESSIONID=11D184014746E006E63083CFADE8AD9D, content-length: 0.

GET http://localhost:8081/idor/info/?requested_customer_id=886359&_=1733567289836 HTTP/1.1

host: localhost:8081

User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:84.0) Gecko/20100101 Firefox/84.0

Accept: application/json, text/javascript, */*; q=0.01

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

X-Requested-With: XMLHttpRequest

Connection: keep-alive

Referer: http://localhost:8081/idor/profile/

Cookie: JSESSIONID=11D184014746E006E63083CFADE8AD9D

content-length: 0

Response body:

Manual Request Editor

RequestResponse

Header: TextBody: TextSend

HTTP/1.1 200
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-Frame-Options: DENY
Content-Type: application/json; charset=UTF-8
Content-Length: 180
Date: Sat, 07 Dec 2024 10:36:24 GMT

Find:

↑ ↓ No matchesTime: 5 msBody Length: 180Total Length: 482 bytes

http://localhost:8081/dor/info/?requested_customer_id=886359&_=17...200

	RTT	Size Resp. ...	Highest Alert
	5 ms	179 bytes	Informational
	23 ms	180 bytes	Informational
	16 ms	4,342 bytes	Medium
	7 ms	121,200 by...	
	8 ms	86,694 bytes	Medium
	24 ms	37,045 bytes	
	2 ms	4,851 bytes	Informational
	11 ms	66 bytes	Informational
	16 ms	3,352 bytes	Medium
	4 ms	121,200 by...	
	23 ms	86,694 bytes	Medium
	5 ms	37,045 bytes	
	2 ms	4,851 bytes	Informational
	3 ms	179 bytes	Informational
	23 ms	179 bytes	Informational
	11 ms	179 bytes	Informational
	11 ms	179 bytes	Informational
	5 ms	180 bytes	Informational

This shows that the server does not validate whether the logged-in user is authorized to view the requested customer's information, resulting in unauthorized data exposure.

Vulnerability assessment and System Assurance Report

13

4.2 IDOR Access Control - Fix

Here is the vulnerable code for the Access Control vulnerability:

```
for (Map.Entry<String, Map<String, String>> entry : customerInfo.entrySet()) {
    String key = entry.getKey();
    Map<String, String> value = entry.getValue();
    String each_customer_id = value.get("id");

    if (requested_customer_id.equals(each_customer_id)) {
        requested_info.put("msg", "You have successfully seen data");
        requested_info.put("name", value.get("name"));
        requested_info.put("email", key);
        requested_info.put("performance", value.get("performance"));
    }
}
```

The issue is that the endpoint is allowing querying of any requested_customer_id without verifying if the logged-in customer was authorized to view the requested data.

And here is the updated code (the main fix is highlighted):

```
@GetMapping("/info")
@ResponseBody
public String idor_performance_info(@RequestParam String requested_customer_id, HttpServletRequest request) throws Exception {
    Map<String, String> requested_info = new HashMap<>();
    HttpSession session = request.getSession();
    String return_val = "";
    try {
        ObjectMapper objectMapper = new ObjectMapper();
        if (session.getAttribute("logged_in_customer") == null) {
            requested_info.put("status", "error");
            requested_info.put("msg", "Please login as customer first");
            return objectMapper.writeValueAsString(requested_info);
        }

        Map<String, String> loggedInCustomer = (Map<String, String>) session.getAttribute("logged_in_customer");
        if (!loggedInCustomer.get("id").equals(requested_customer_id)) {
            requested_info.put("status", "error");
            requested_info.put("msg", "You are not authorized to view this data");
            return objectMapper.writeValueAsString(requested_info);
        }

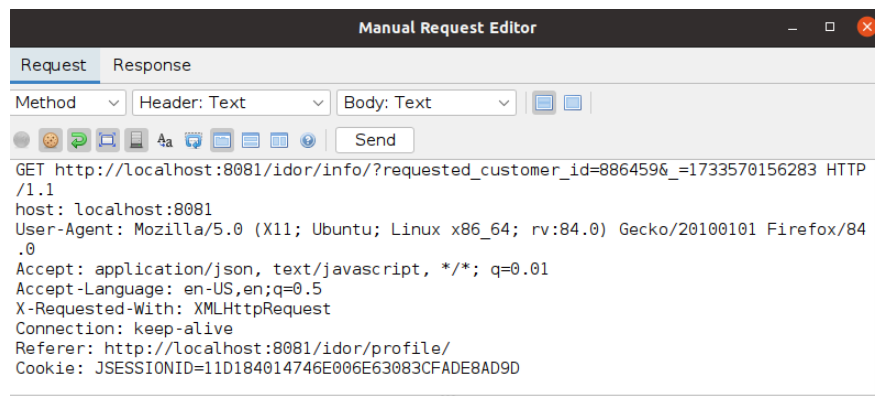
        for (Map.Entry<String, Map<String, String>> entry : customerInfo.entrySet()) {
            String key = entry.getKey();
            Map<String, String> value = entry.getValue();
            String each_customer_id = value.get("id");

            if (requested_customer_id.equals(each_customer_id)) {
                requested_info.put("msg", "You have successfully seen data");
                requested_info.put("name", value.get("name"));
                requested_info.put("email", key);
                requested_info.put("performance", value.get("performance"));
            }
        }
        return_val = objectMapper.writeValueAsString(requested_info);
    } catch (JsonProcessingException e) {
        e.printStackTrace();
        return_val = "{\"status\":\"error\"}";
    }
    return return_val;
}
```

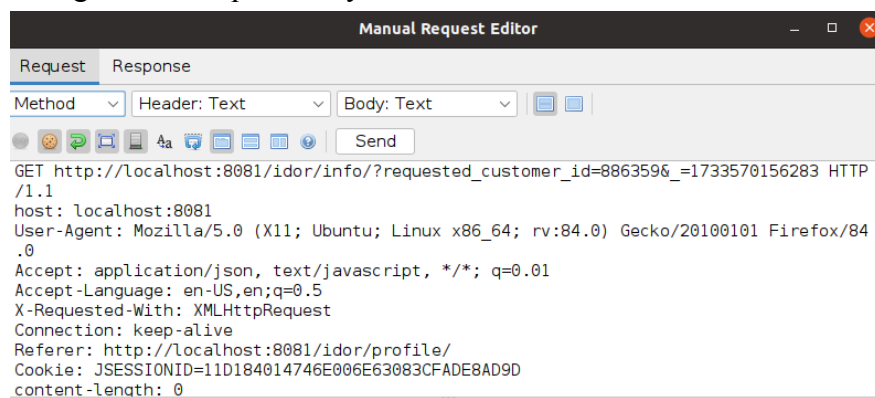
This checks if the logged-in customer's ID (loggedInCustomer.get("id")) matches the requested customer ID (requested_customer_id). If the IDs do not match, it returns an error message stating that the user is not authorized to view the requested data.

Proof of it working can be shown on the next page:

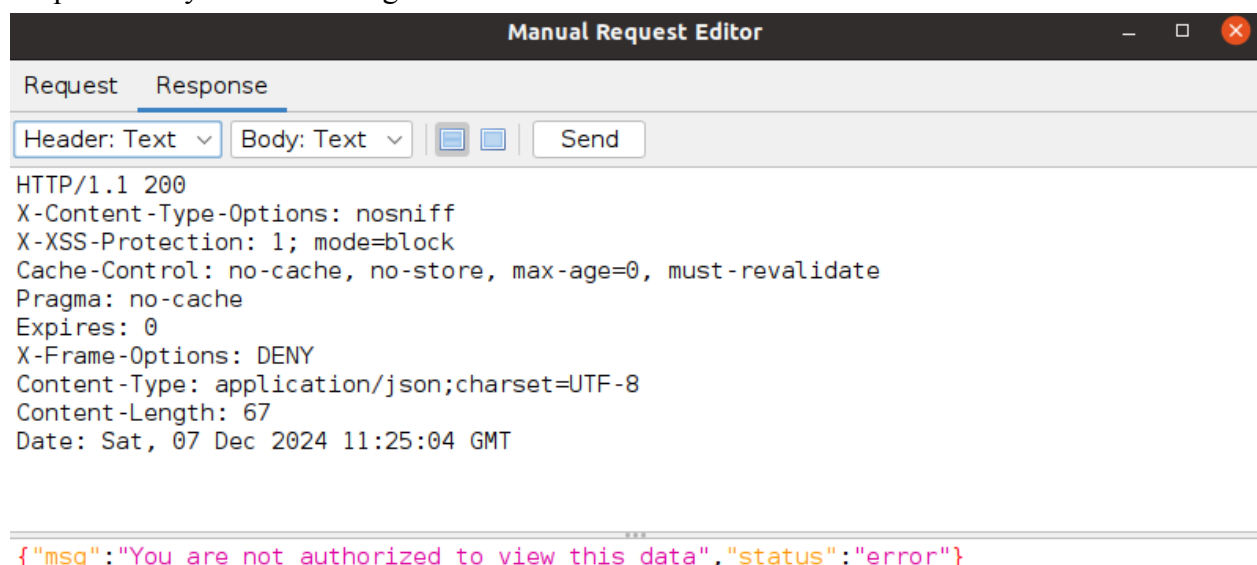
Original GET request body:



Changed GET request body:



Response body after the change:

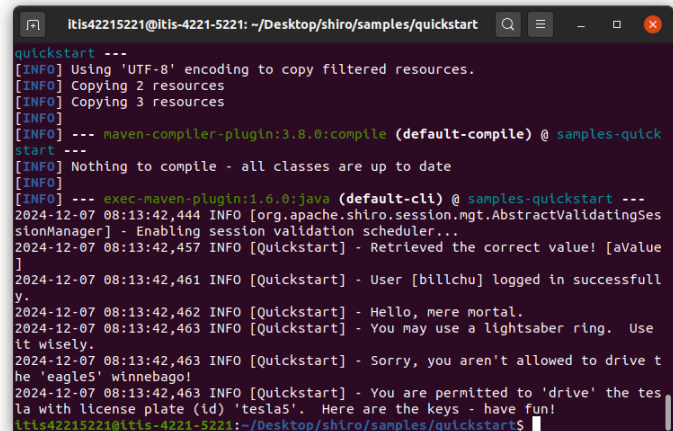


5.0 Apache Shiro Access Control File

5.1 Apache Shiro Access Control File - Before Fix

Here is the access control code before the fix and the result of running the original code:

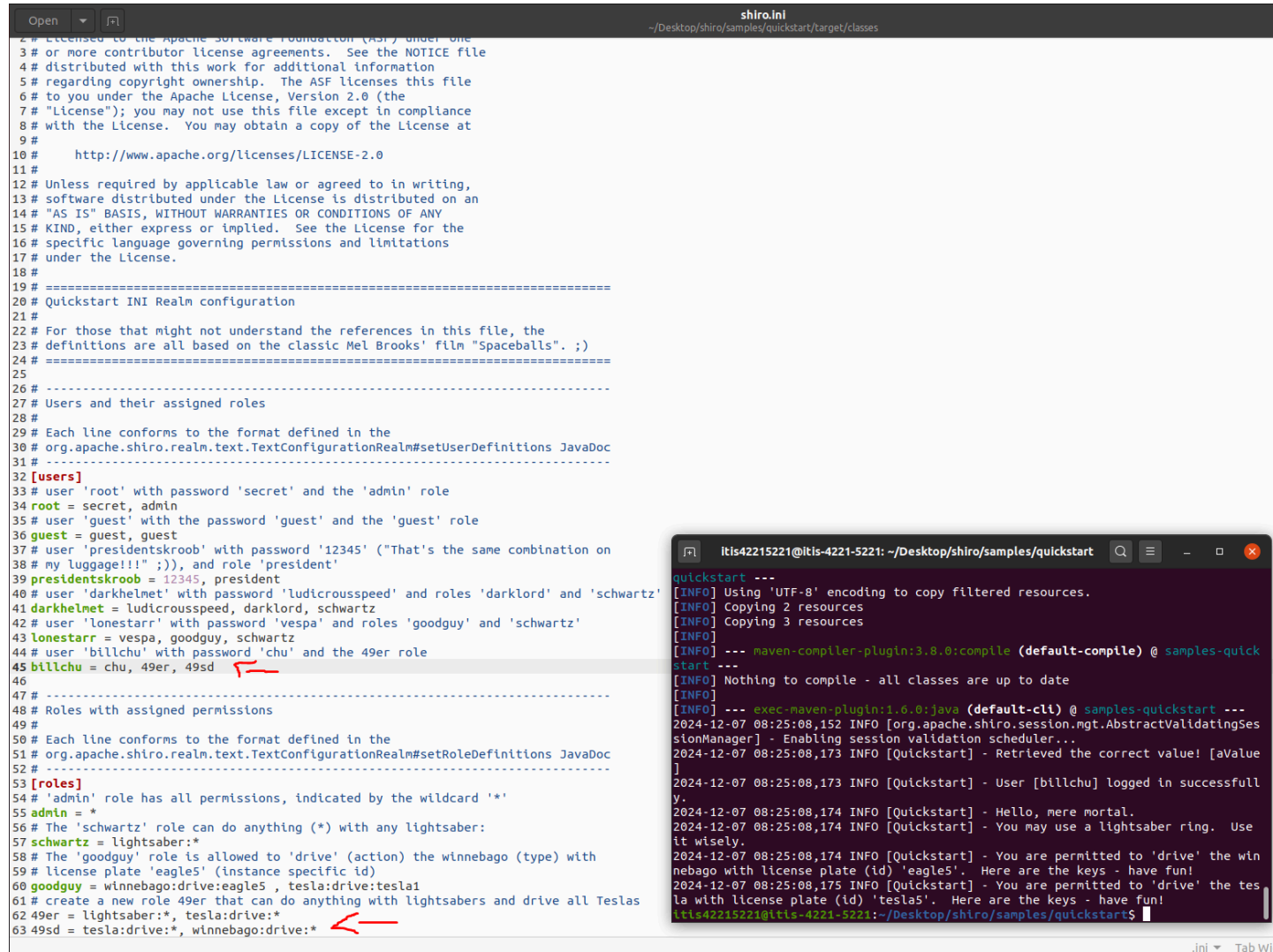
```
3# or more contributor license agreements. See the NOTICE file
4# distributed with this work for additional information
5# regarding copyright ownership. The ASF licenses this file
6# to you under the Apache License, Version 2.0 (the
7# "License"); you may not use this file except in compliance
8# with the License. You may obtain a copy of the License at
9#
10# http://www.apache.org/licenses/LICENSE-2.0
11#
12# Unless required by applicable law or agreed to in writing,
13# software distributed under the License is distributed on an
14# "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
15# KIND, either express or implied. See the License for the
16# specific language governing permissions and limitations
17# under the License.
18#
19# =====
20# Quickstart INI Realm configuration
21#
22# For those that might not understand the references in this file, the
23# definitions are all based on the classic Mel Brooks' film "Spaceballs". ;)
24# =====
25#
26# -----
27# Users and their assigned roles
28#
29# Each line conforms to the format defined in the
30# org.apache.shiro.realm.text.TextConfigurationRealm#setUserDefinitions JavaDoc
31# -----
32# [users]
33# user 'root' with password 'secret' and the 'admin' role
34# root = secret, admin
35# user 'guest' with the password 'guest' and the 'guest' role
36# guest = guest, guest
37# user 'presidentskroob' with password '12345' ("That's the same combination on
38# my luggage!!" ;)), and role 'president'
39# presidentskroob = 12345, president
40# user 'darkhelmet' with password 'ludicrouspeed' and roles 'darklord' and 'schwartz'
41# darkhelmet = ludicrousspeed, darklord, schwartz
42# user 'lonestarr' with password 'vespa' and roles 'goodguy' and 'schwartz'
43# lonestarr = vespa, goodguy, schwartz
44# user 'billchu' with password 'chu' and the 49er role
45# billchu = chu, 49er
46#
47# -----
48# Roles with assigned permissions
49#
50# Each line conforms to the format defined in the
51# org.apache.shiro.realm.text.TextConfigurationRealm#setRoleDefinitions JavaDoc
52# -----
53# [roles]
54# 'admin' role has all permissions, indicated by the wildcard '*'
55# admin = *
56# The 'schwartz' role can do anything (*) with any lightsaber:
57# schwartz = lightsaber:*
58# The 'goodguy' role is allowed to 'drive' (action) the winnebago (type) with
59# license plate 'eagle5' (instance specific id)
60# goodguy = winnebago:drive:eagle5, tesla:drive:tesla1
61# create a new role 49er that can do anything with lightsabers and drive all Teslas
62# 49er = lightsaber:*, tesla:drive:*
```



```
Itis42215221@Itis-4221-5221: ~/Desktop/shiro/samples/quickstart
quickstart ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Copying 2 resources
[INFO] Copying 3 resources
[INFO] --- maven-compiler-plugin:3.8.0:compile (default-compile) @ samples-quickstart ---
[INFO] Nothing to compile - all classes are up to date
[INFO] --- exec-maven-plugin:1.6.0:java (default-cli) @ samples-quickstart ---
2024-12-07 08:13:42,444 INFO [org.apache.shiro.session.mgt.AbstractValidatingSessionManager] - Enabling session validation scheduler...
2024-12-07 08:13:42,457 INFO [Quickstart] - Retrieved the correct value! [aValue]
2024-12-07 08:13:42,461 INFO [Quickstart] - User [billchu] logged in successfully.
2024-12-07 08:13:42,462 INFO [Quickstart] - Hello, mere mortal.
2024-12-07 08:13:42,463 INFO [Quickstart] - You may use a lightsaber ring. Use it wisely.
2024-12-07 08:13:42,463 INFO [Quickstart] - Sorry, you aren't allowed to drive the 'eagle5' winnebago!
2024-12-07 08:13:42,463 INFO [Quickstart] - You are permitted to 'drive' the tesla with license plate (id) 'tesla5'. Here are the keys - have fun!
Itis42215221@Itis-4221-5221:~/Desktop/shiro/samples/quickstart$
```


5.2 Apache Shiro Access Control File - After Fix

Here is the access control code and the result of running it after the new roles and permissions were added:



```
Open shiro.ini
~/Desktop/shiro/samples/quickstart/target/classes

2 # Licensed to the Apache Software Foundation (ASF) under one
3 # or more contributor license agreements. See the NOTICE file
4 # distributed with this work for additional information
5 # regarding copyright ownership. The ASF licenses this file
6 # to you under the Apache License, Version 2.0 (the
7 # "License"); you may not use this file except in compliance
8 # with the License. You may obtain a copy of the License at
9 #
10 # http://www.apache.org/licenses/LICENSE-2.0
11 #
12 # Unless required by applicable law or agreed to in writing,
13 # software distributed under the License is distributed on an
14 # "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
15 # KIND, either express or implied. See the License for the
16 # specific language governing permissions and limitations
17 # under the License.
18 #
19 # =====
20 # Quickstart INI Realm configuration
21 #
22 # For those that might not understand the references in this file, the
23 # definitions are all based on the classic Mel Brooks' film "Spaceballs". ;)
24 # =====
25 #
26 # -----
27 # Users and their assigned roles
28 #
29 # Each line conforms to the format defined in the
30 # org.apache.shiro.realm.text.TextConfigurationRealm#setUserDefinitions JavaDoc
31 # -----
32 [users]
33 # user 'root' with password 'secret' and the 'admin' role
34 root = secret, admin
35 # user 'guest' with the password 'guest' and the 'guest' role
36 guest = guest, guest
37 # user 'presidentskroob' with password '12345' ("That's the same combination on
38 # my luggage!!!";)), and role 'president'
39 presidentskroob = 12345, president
40 # user 'darkhelmet' with password 'ludicrousspeed' and roles 'darklord' and 'schwartz'
41 darkhelmet = ludicrousspeed, darklord, schwartz
42 # user 'lonestarr' with password 'vespa' and roles 'goodguy' and 'schwartz'
43 lonestarr = vespa, goodguy, schwartz
44 # user 'billchu' with password 'chu' and the 49er role
45 billchu = chu, 49er, 49sd
46 #
47 # -----
48 # Roles with assigned permissions
49 #
50 # Each line conforms to the format defined in the
51 # org.apache.shiro.realm.text.TextConfigurationRealm#setRoleDefinitions JavaDoc
52 # -----
53 [roles]
54 # 'admin' role has all permissions, indicated by the wildcard '*'
55 admin = *
56 # The 'schwartz' role can do anything (*) with any lightsaber:
57 schwartz = lightsaber:*
58 # The 'goodguy' role is allowed to 'drive' (action) the winnebago (type) with
59 # license plate 'eagle5' (instance specific id)
60 goodguy = winnebago:drive:eagle5, tesla:drive:tesla1
61 # create a new role 49er that can do anything with lightsabers and drive all Teslas
62 49er = lightsaber:*, tesla:drive:*
63 49sd = tesla:drive:*, winnebago:drive:*
```

```
quickstart ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Copying 2 resources
[INFO] Copying 3 resources
[INFO] --- maven-compiler-plugin:3.8.0:compile (default-compile) @ samples-quickstart ---
[INFO] Nothing to compile - all classes are up to date
[INFO] --- exec-maven-plugin:1.6.0:java (default-cli) @ samples-quickstart ---
2024-12-07 08:25:08,152 INFO [org.apache.shiro.session.mgt.AbstractValidatingSessionManager] - Enabling session validation scheduler...
2024-12-07 08:25:08,173 INFO [Quickstart] - Retrieved the correct value! [aValue]
2024-12-07 08:25:08,173 INFO [Quickstart] - User [billchu] logged in successfully.
2024-12-07 08:25:08,174 INFO [Quickstart] - Hello, mere mortal.
2024-12-07 08:25:08,174 INFO [Quickstart] - You may use a lightsaber ring. Use it wisely.
2024-12-07 08:25:08,174 INFO [Quickstart] - You are permitted to 'drive' the winnebago with license plate (id) 'eagle5'. Here are the keys - have fun!
2024-12-07 08:25:08,175 INFO [Quickstart] - You are permitted to 'drive' the tesla with license plate (id) 'tesla5'. Here are the keys - have fun!
itls42215221@itls-4221-5221: ~/Desktop/shiro/samples/quickstart$
```