SYSLIFTERS

Jedox Vulnerability Disclosure

Vendor: **Jedox GmbH**
2023-01-18

Contact:
hello@syslifters.com

Content

Our Disclosure Policy

Syslifters is committed to responsibly disclosing security vulnerabilities to product vendors and the general public. We believe that responsible disclosure is a shared responsibility between vendors and researchers alike and the best way to ensure security for users. By adhering to this policy, we maintain transparency and can work together to make the internet a safer place for everyone.

To ensure that vulnerabilities are addressed in a timely manner, we adhere to a 90-day disclosure deadline and reserve the right to directly notify affected clients. We immediately notify vendors of vulnerabilities through any appropriate contacts or formal mechanisms listed on the vendor's website, with details shared with the public after 90 days, or sooner if a fix is released by the vendor.

This policy is subject to the following exceptions:

- If a vendor informs us that a patch is scheduled for release within 14 days following the deadline, we will delay the public disclosure until the availability of the patch.
- In cases of zero-day vulnerabilities (vulnerabilities that are actively being exploited), we believe that more urgent action is necessary, and we will disclose the vulnerability within 7 days. This is because each day that a vulnerability remains undisclosed and unpatched, more devices or accounts are at risk of being compromised.
- We reserve the right to adjust the 90-day disclosure deadline in accordance with circumstances deemed reasonable by the vendor. We remain committed to treating all vendors equally and expect the same standard to be applied to us.

Here is the report. What now?

Our initial contact with your security staff was on 20 December 2022. Following our responsible disclosure policy, we plan to publicly release the vulnerability information in mid-April. We may adjust the disclosure deadline in accordance with circumstances deemed reasonable.

We have requested corresponding CVE numbers for you and have already pre-filled details on the vulnerabilities. CVE records can be updated at any time via CVE.org.

For support, we are of course also available for a personal exchange. If you have any further questions you can contact us at any time via the corresponding contact options.

Identified Vulnerabilities

5 Critical, **1 High** and **1 Medium** vulnerabilities were identified:

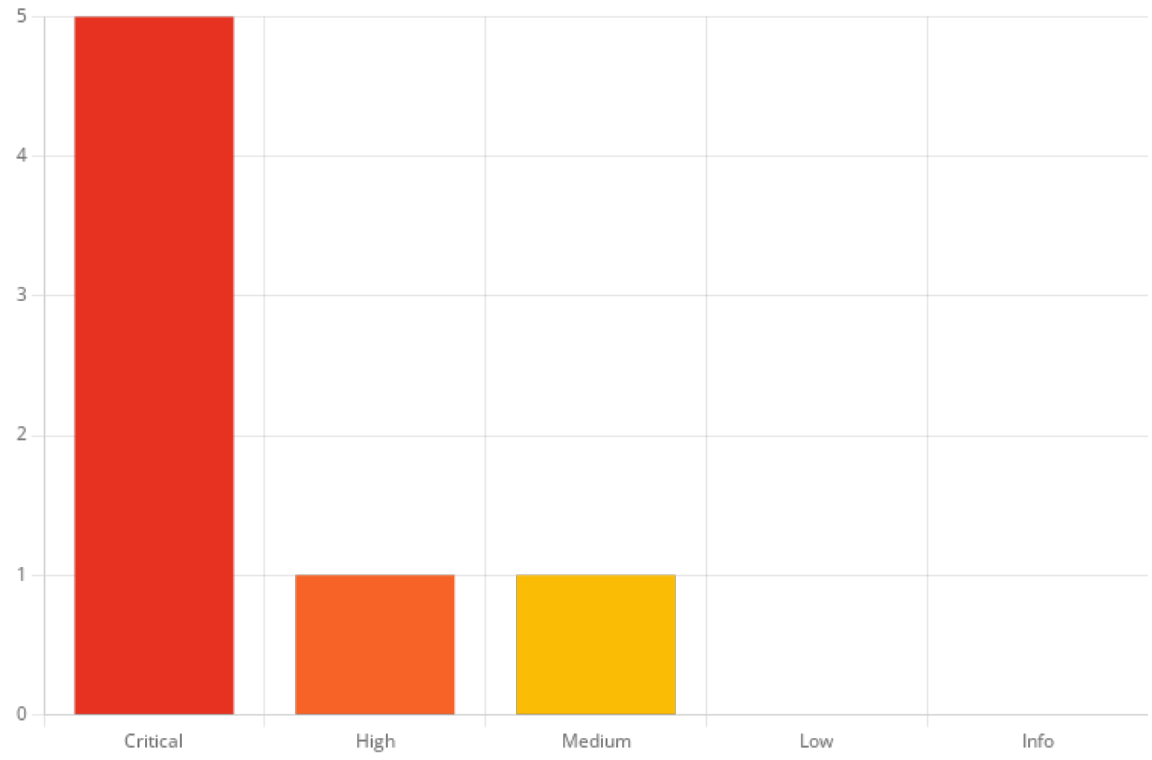

Figure 1 - Distribution of identified vulnerabilities

The following vulnerabilities were found:

Type	Vulnerability	Criticality
Directory Traversal	CVE-2022-47875: Remote Code Execution via Directory Traversal	**Critical**
Exec Code	CVE-2022-47879: Code Execution via RPC Interfaces	**Critical**
Cross Site Scripting	CVE-2022-47877: Stored Cross-Site Scripting in Log-Module	**Critical**
Exec Code	CVE-2022-47876: Remote Code Execution via Executable Groovy-Scripts	**Critical**
Incorrect Input Validation	CVE-2022-47878: Remote Code Execution via Configurable Storage Path	**Critical**

Type	Vulnerability	Criticality
Incorrect Access Control	CVE-2022-47874: Disclosure of Database Credentials via Improper Access Controls	High
Information Disclosure	CVE-2022-47880: Disclosure of Database Credentials via Connection Checks	Medium

CVE-2022-47875: Remote Code Execution via Directory Traversal

Criticality: Critical
CVSS-Score: 9.9
Vulnerability Type: Directory Traversal
Product:
- Jedox Cloud
- Jedox 2020.2.5

Affects: Component: /be/erpc.php

Overview

A Directory Traversal vulnerability in */be/erpc.php* in Jedox Cloud and Jedox 2020.2.5 allows remote authenticated users to execute arbitrary code. To exploit the vulnerability, the attacker must have the permissions to upload files.

Description

We were able to identify an RPC functionality `erpc.php` on the affected web server. Using this function, arbitrary methods of already loaded program classes can be executed. This already poses a significant security risk, since many methods do not perform any authorization checks for potentially dangerous functions (see also CVE-2022-47879).

If a program class is not loaded, the application tries to load the class from the directory `rtn`:

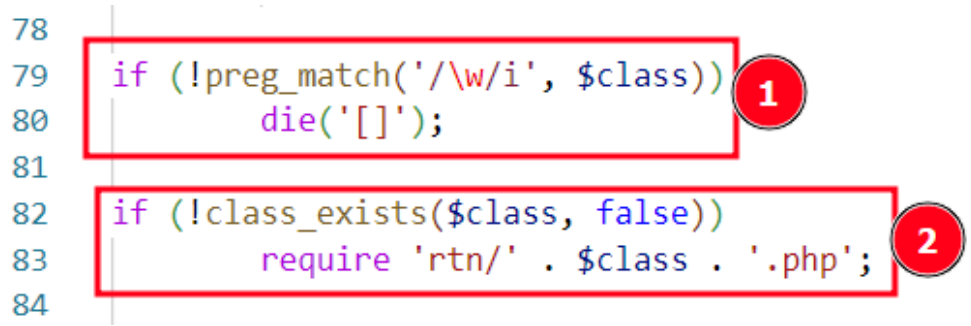

Figure 2 - Excerpt from the program code of the file erpc.php, licensed under GPL 2, Copyright (C) 2006-2010 Jedox AG

As can be seen in the excerpt from the program code (from an older Open Source version suffering from the same issues) under point 1, the name of the class is validated against a regex, which is does not validate the user supplied value properly. The only requirement for the entered value is the presence of a letter, number or underscore in any position.

Point 2 marks the path of the class to be loaded which is composed with the non-validated variable `class`.

Therefore, it is possible to load arbitrary PHP files from the file system using a relative path such as `../../../storage/h1-Public+Files/n42-pentest`.

An attacker can now abuse this vulnerability by first uploading a file using one of the existing file upload functions, as shown in the following screenshot:

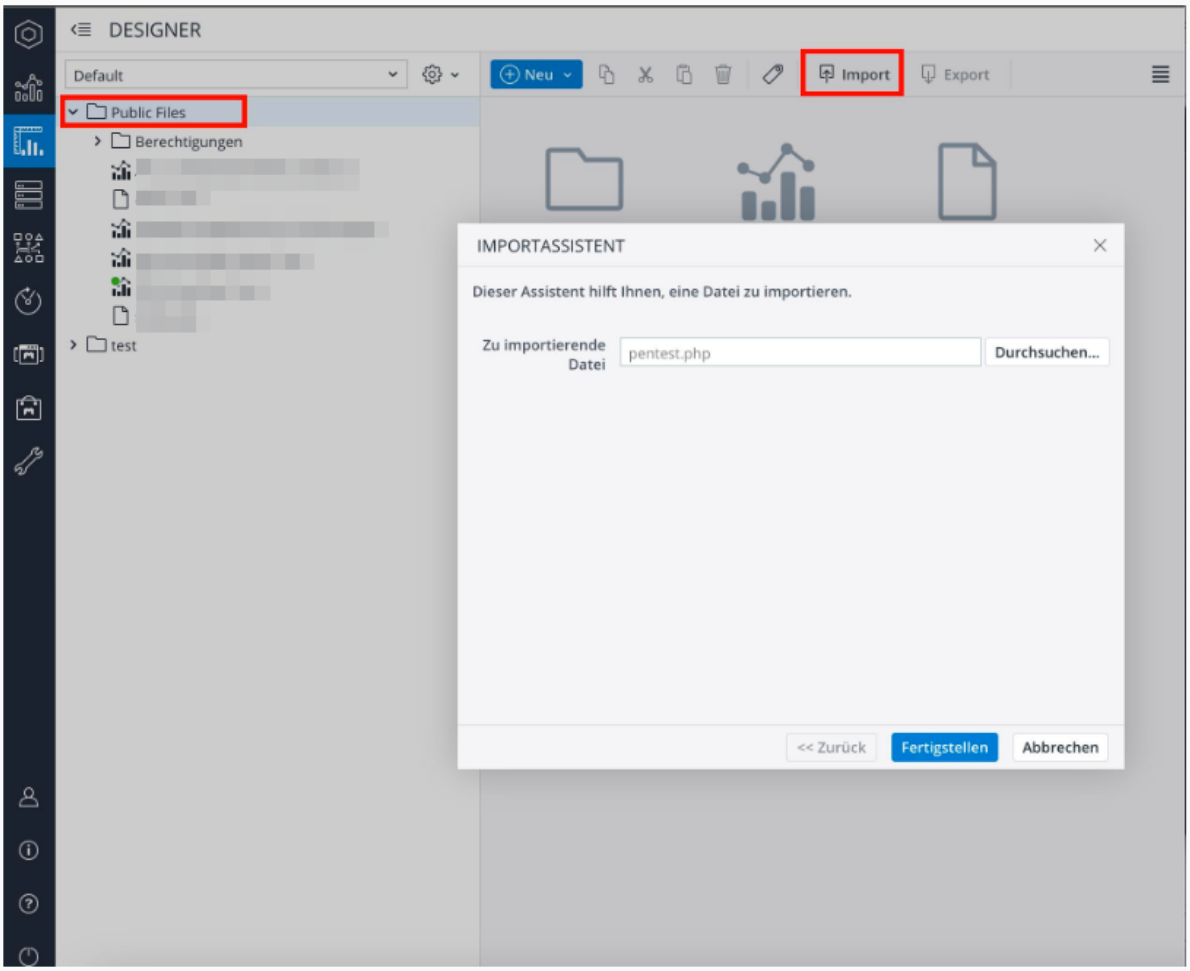

Figure 3 - Upload a file

When uploading a file, the web application returns the location on the file system of the uploaded file:

```
{
  "success":true,
  "log":null,
  "nodeId":"n43",
  "name":"pentest",
  "fspath":"\/storage\/h1-Public+Files\/n43-pentest.php.php",
  "path":"\/\/Default\/Public Files\/pentest.php",
  "g":"fgrp1",
  "h":"h1",
  "n":"n43"
}
```


Figure 4 - Response to uploading a file

It is now possible to exploit the vulnerability and execute the malicious file:

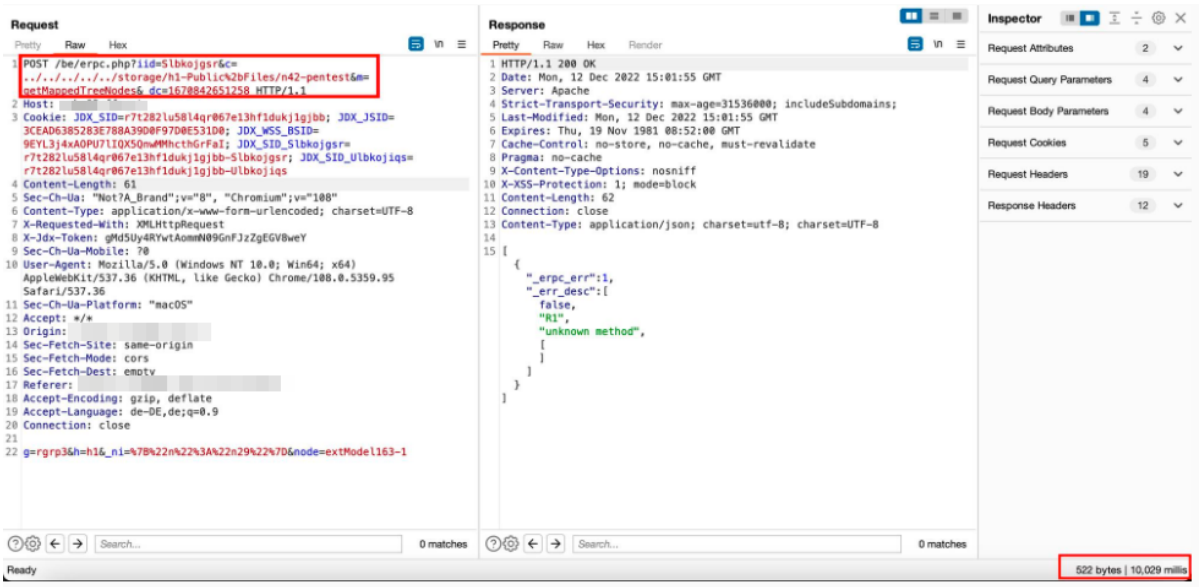

Figure 5 - Remote Code Execution (RCE)

The preceding screenshot shows that the uploaded file has been executed. For demo purposes, the malicious payload will only make the server wait for 10-second before it responds.

CVE-2022-47879: Code Execution via RPC Interfaces

Criticality: Critical
CVSS-Score: 9.9
Vulnerability Type: Exec Code
Product:
- Jedox Cloud
- Jedox 2020.2.5

Affects:
- Component: /be/rpc.php
- Component: /be/erpc.php

Overview

A Remote Code Execution (RCE) vulnerability in */be/rpc.php* and */be/erpc.php* in Jedox Cloud and Jedox 2020.2.5 allows remote authenticated users to load arbitrary PHP classes from the *rtn* directory and to execute its methods. To exploit this vulnerability, the attacker needs knowledge about loadable classes, their methods and arguments.

Description

Many functions accessible via multiple RPC interfaces were identified that disclose sensitive data or lead to vulnerabilities. All specified functions can also be executed with read-only user permissions.

The `Studio::getUserCreds` function can be used to read the clear text credentials of the currently authenticated user. This is especially critical in combination with the existing Stored XSS vulnerability (see CVE-2022-47877), as this allows an anonymous (auto-login) user to steal an administrator's password in clear text:

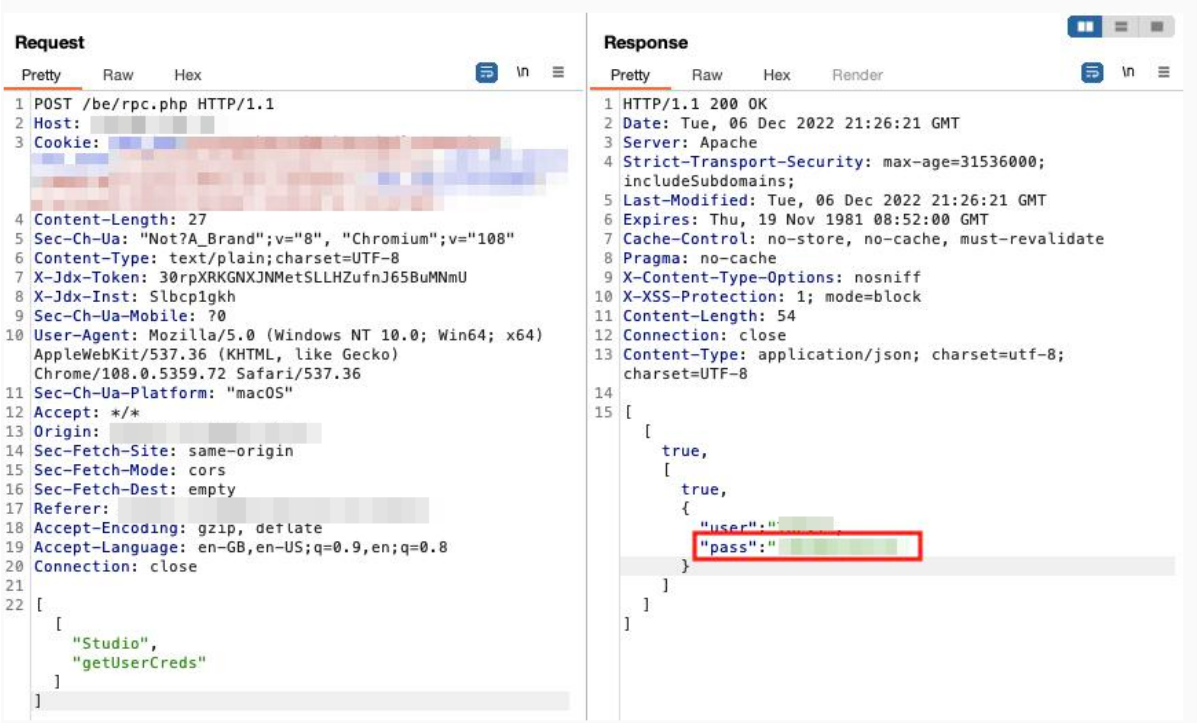

Figure 6 - `Studio::getUserCreds`

Using function `conn::test_palo`, an outgoing HTTP connection can be initiated from the web server with the authenticated user's credentials. This could leak cleartext credentials to an attacker:

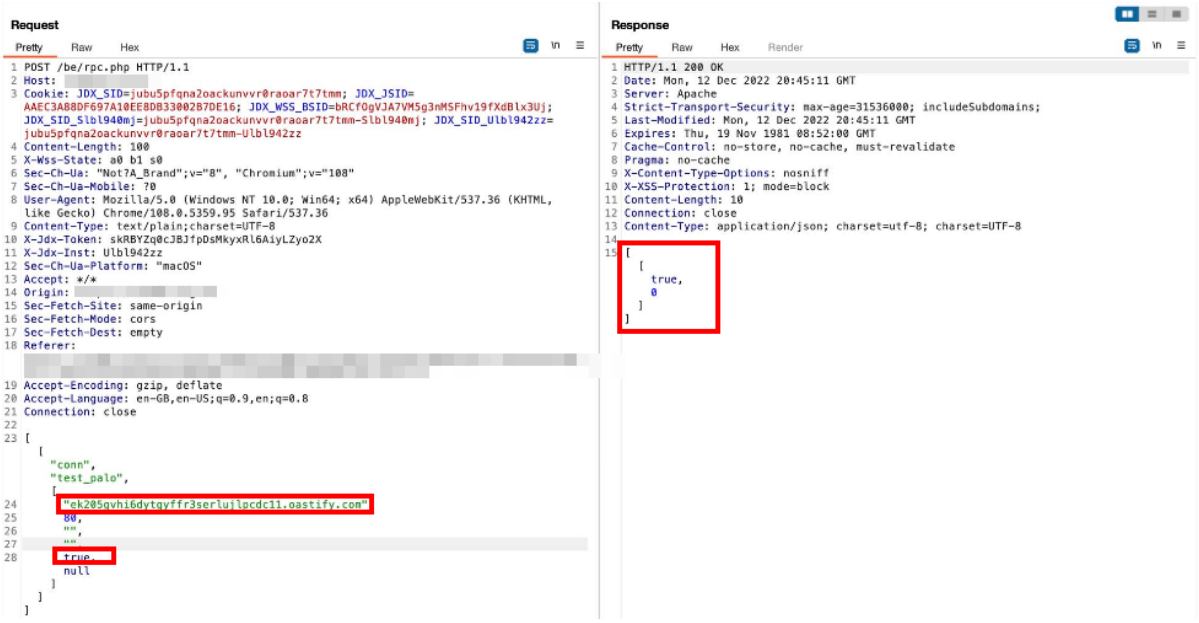

Figure 7 - `conn::test_palo`

As shown in the following screenshot, the server accesses the address via HTTP and transmits the username and password of the logged-in user.

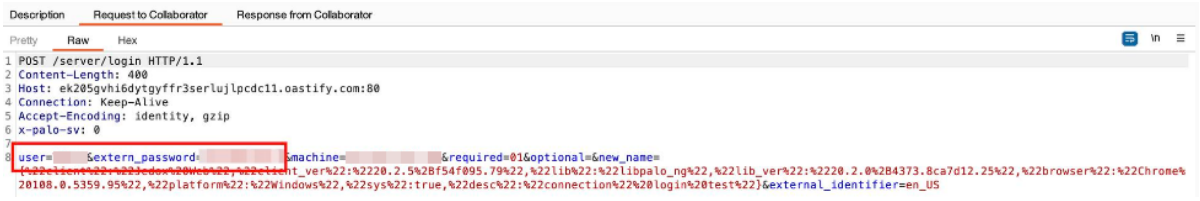

Figure 8 - Capturing cleartext credentials

This method is very suitable for an attack using a XSS vulnerability. The password does not have to be exfiltrated by bypassing cross-origin protection mechanisms, but is sent from the server to the attacker's system.

The function `Studio::getExternalUrl` can be used to generate a URL with embedded username and password.

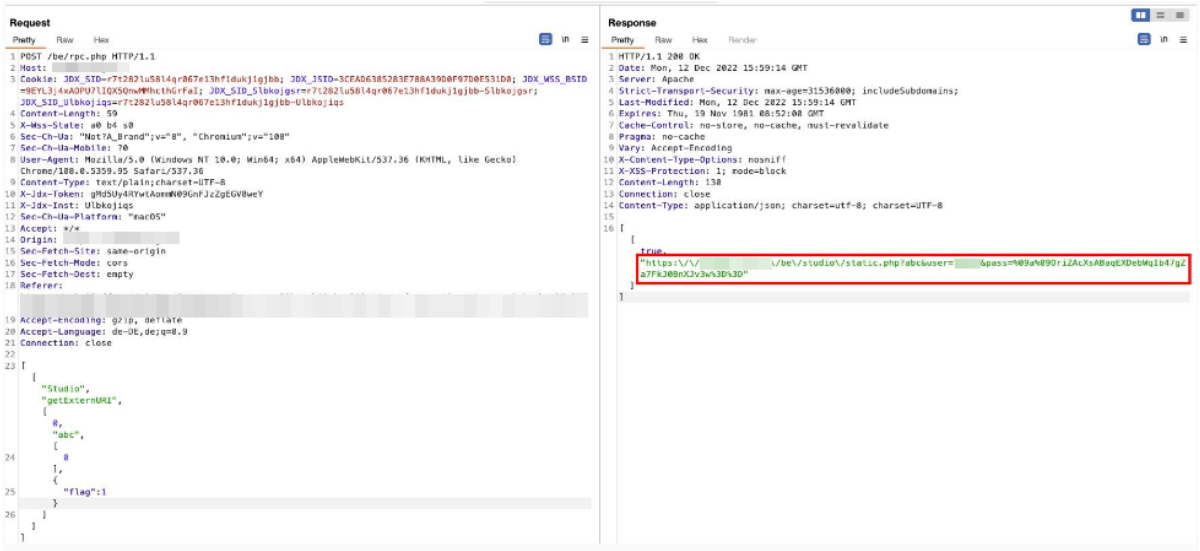

Figure 9 - `Studio->getExternalUrl`

A list of all database connections can be retrieved via `conn::ls`:

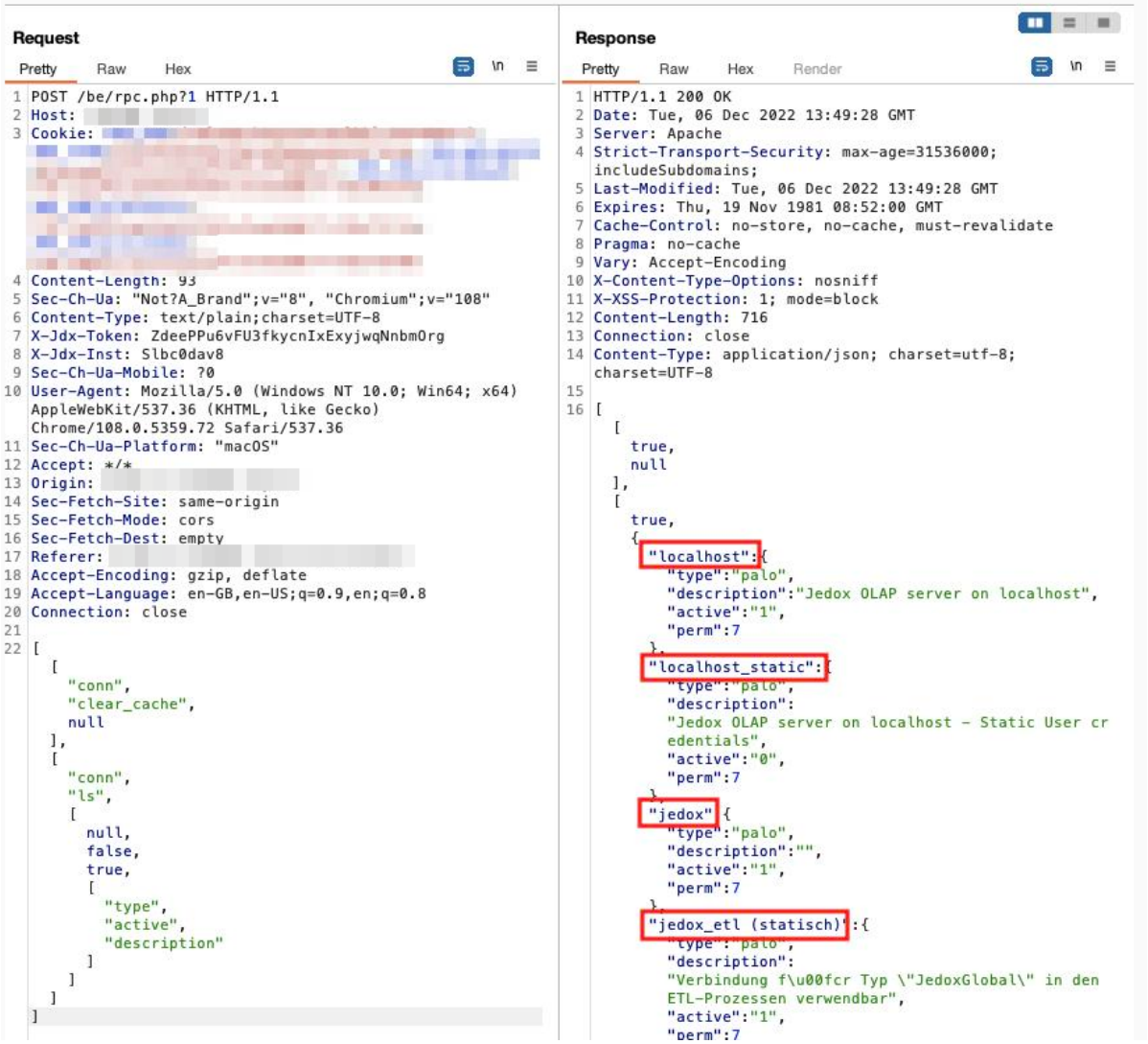

Figure 10 - Query the list of all db connections

Details of individual database connections (including encrypted credentials) can be retrieved using the Java RPC function `com.jedox.etl.mngr.Connection::getGlobalConnection`.

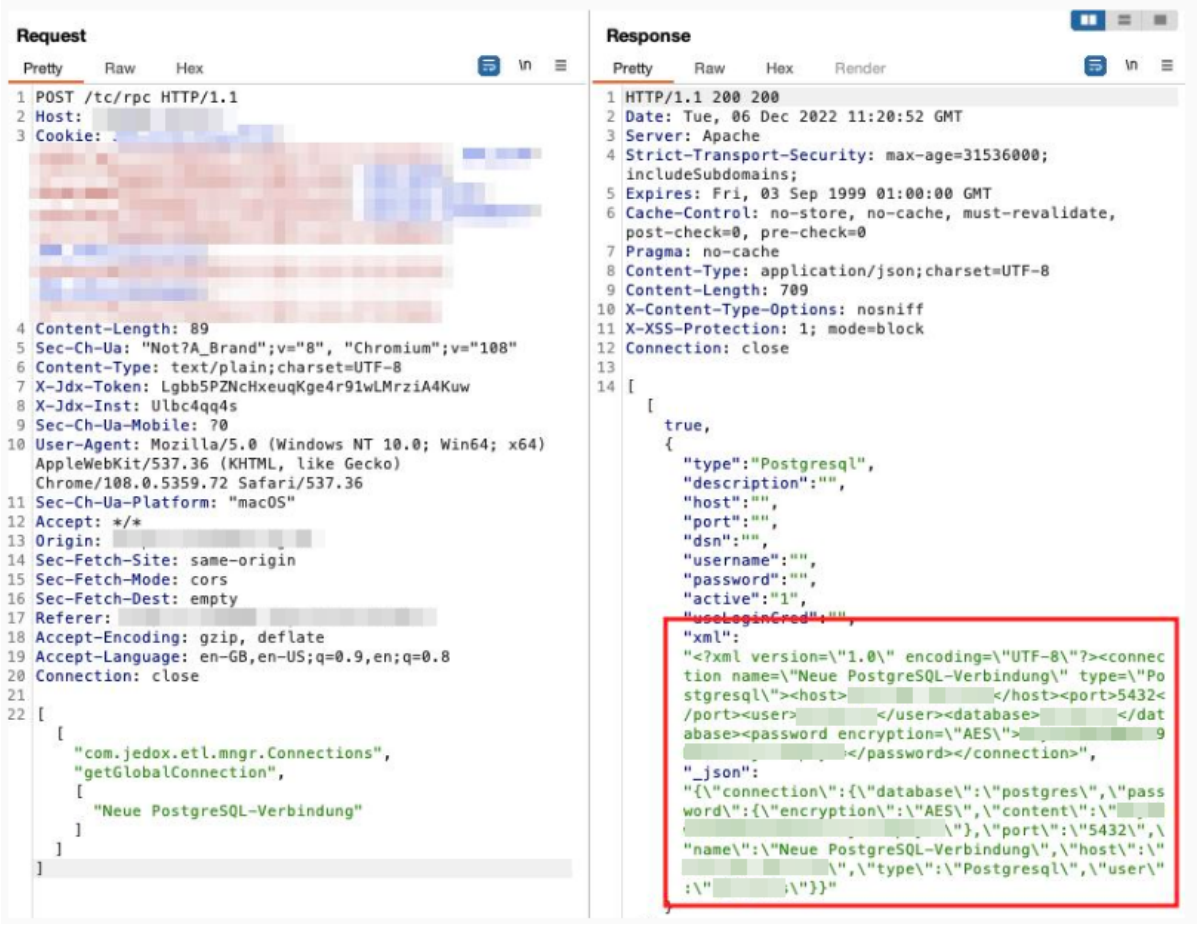

Figure 11 - Retrieve connection details

Some functions return credentials only in encrypted form. However, they can be decrypted by any user using `common::decrypt`:

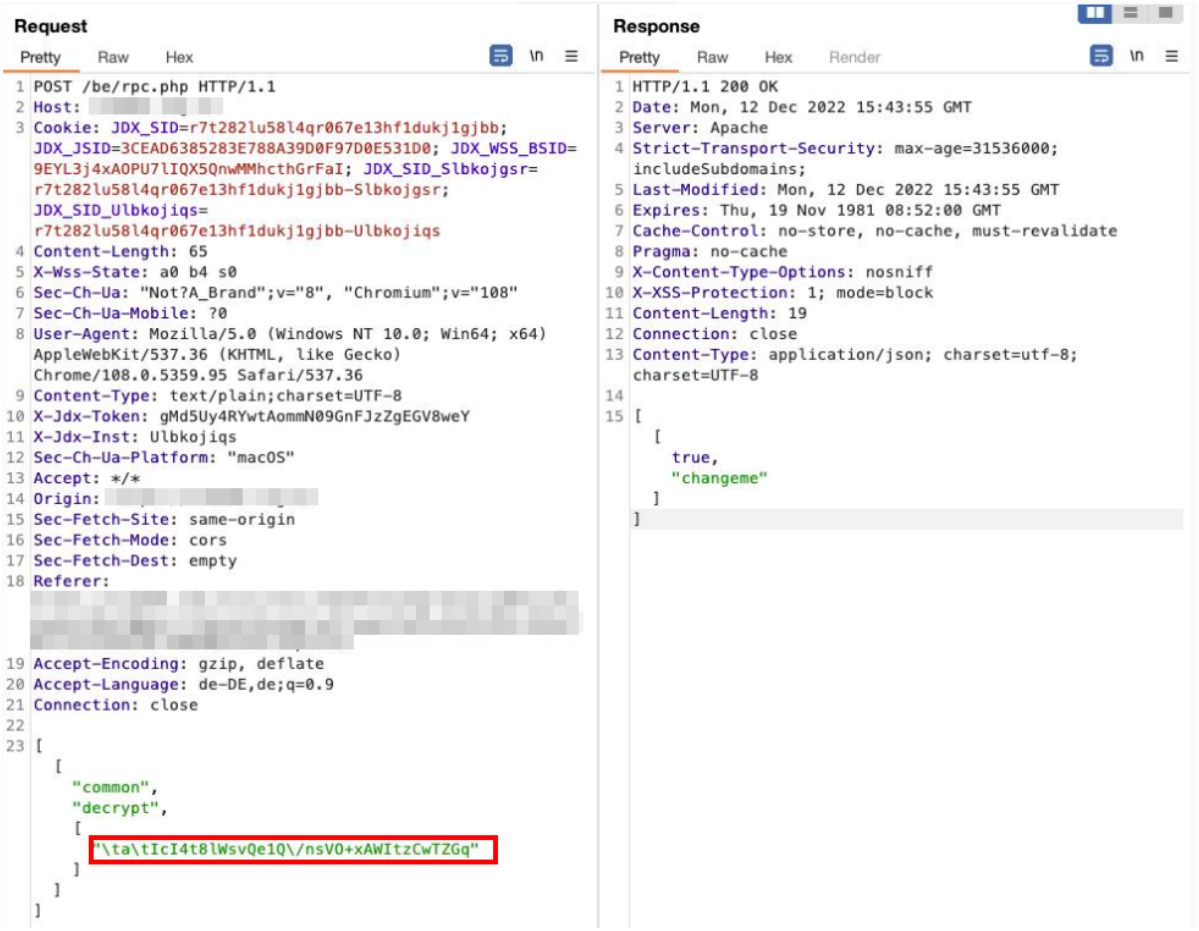

Figure 12 - `common->decrypt`

Using `common::paloGet` it is also possible to read arbitrary configuration parameters. For example, the password of the SMTP server can be read with it:

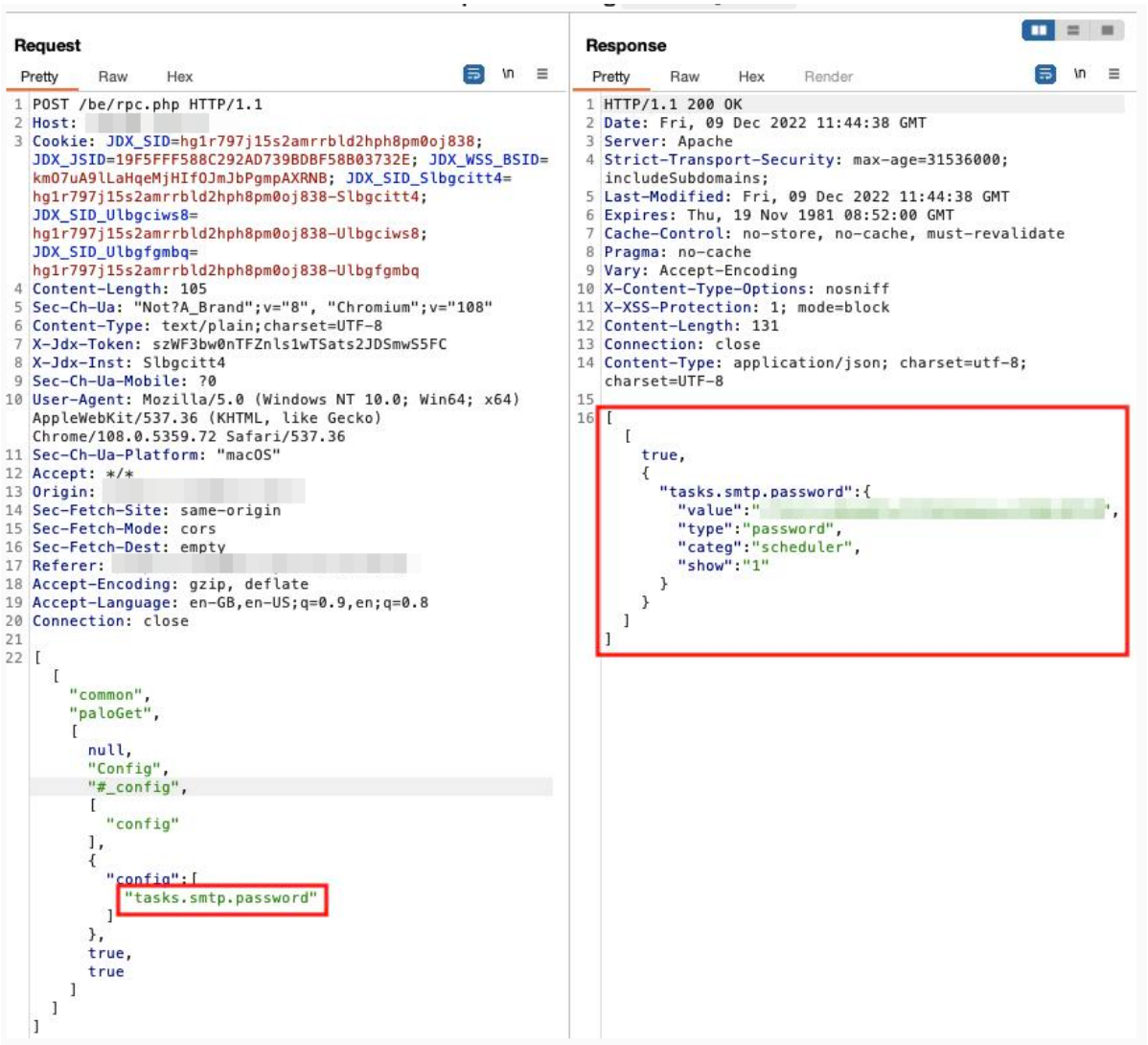

Figure 13 - Read SMTP password using `common::paloGet`

The function `palo_mgmt::sess_list` can be used to retrieve a list of all active user sessions. The session information includes not only the username but also the user's IP address, information about the browser and other data.

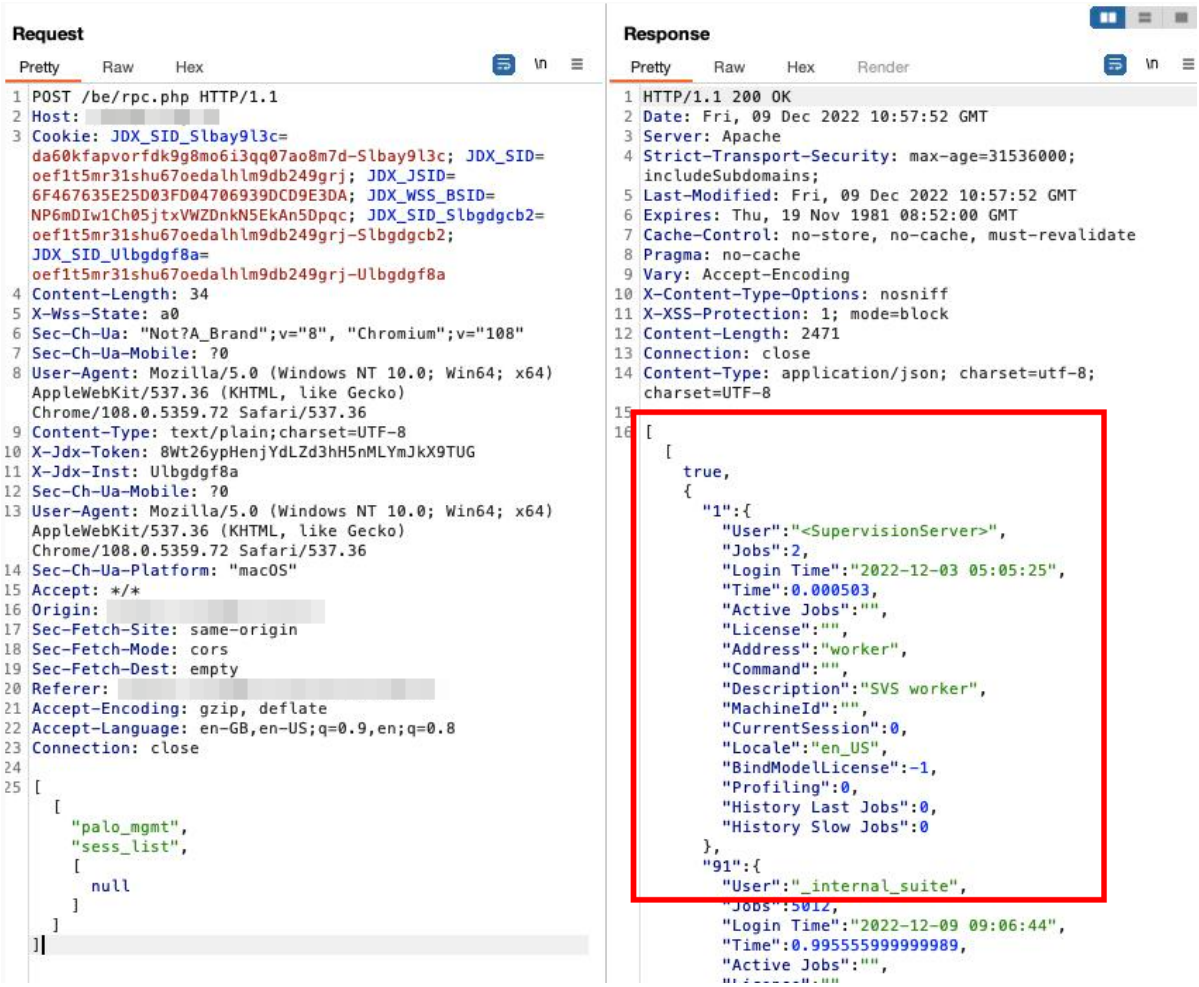

Figure 14 - `palo_mgmt::sess_list`

The function `palo_mgmt::lic_users_list` returns a list of all users stored in the system:

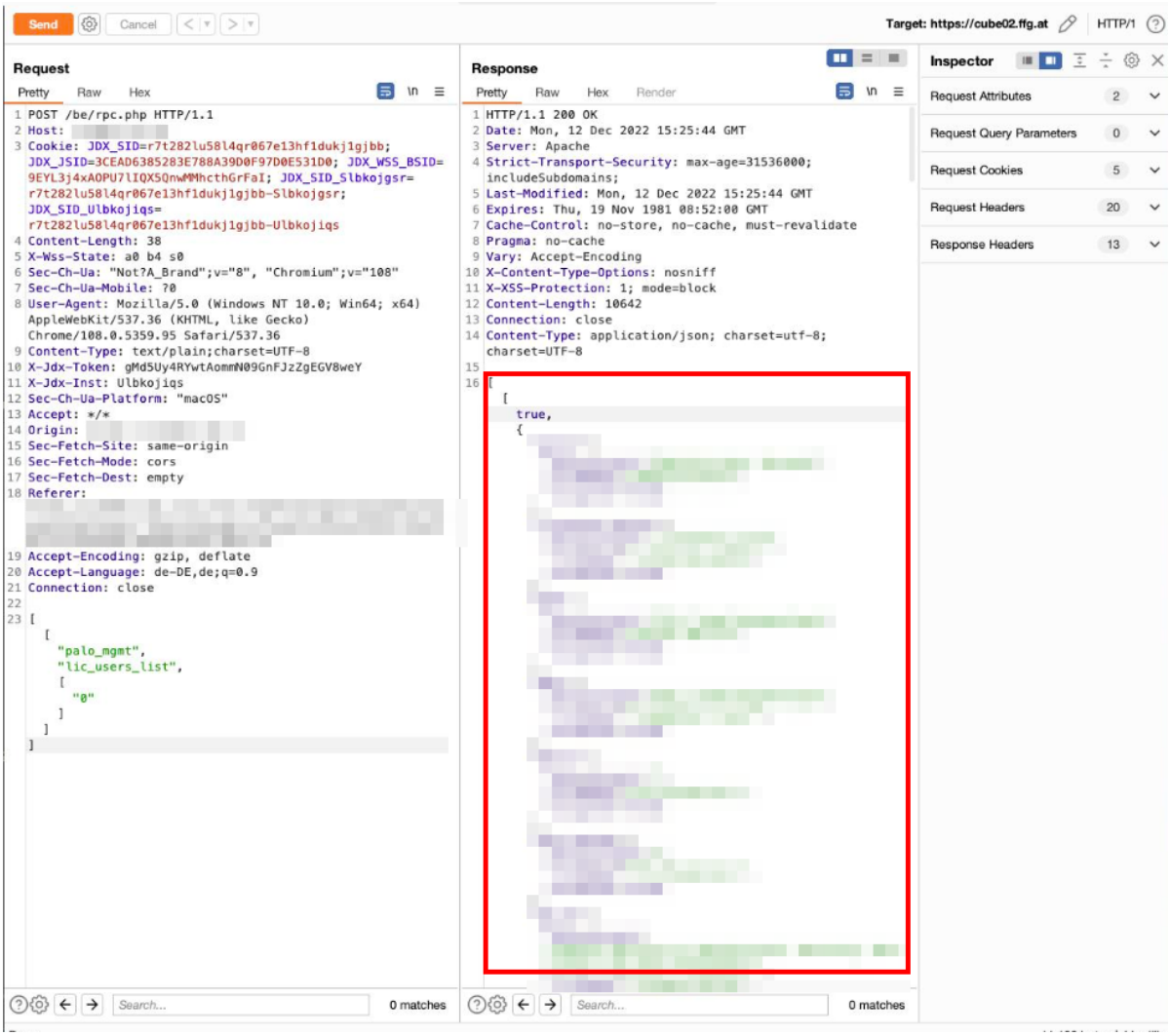

Figure 15 - `palo_mgmt::lic_users_list`

CVE-2022-47877: Stored Cross-Site Scripting in Log-Module

Criticality: Critical
CVSS-Score: 9.6
Vulnerability Type: Cross Site Scripting
Product: Jedox 2020.2.5
Affects:
- Component: /ub/ccmd
- Logs page

Overview

A Stored cross-site scripting vulnerability in Jedox 2020.2.5 allows remote authenticated users to inject arbitrary web scripts or HTML in the logs page via the log module. To exploit the vulnerability, the attacker must append an XSS payload to the log message.

Description

A persistent XSS vulnerability has been identified in the Jedox web application when storing log entries. The following code triggers a demo XSS payload demonstrating the feasibility:

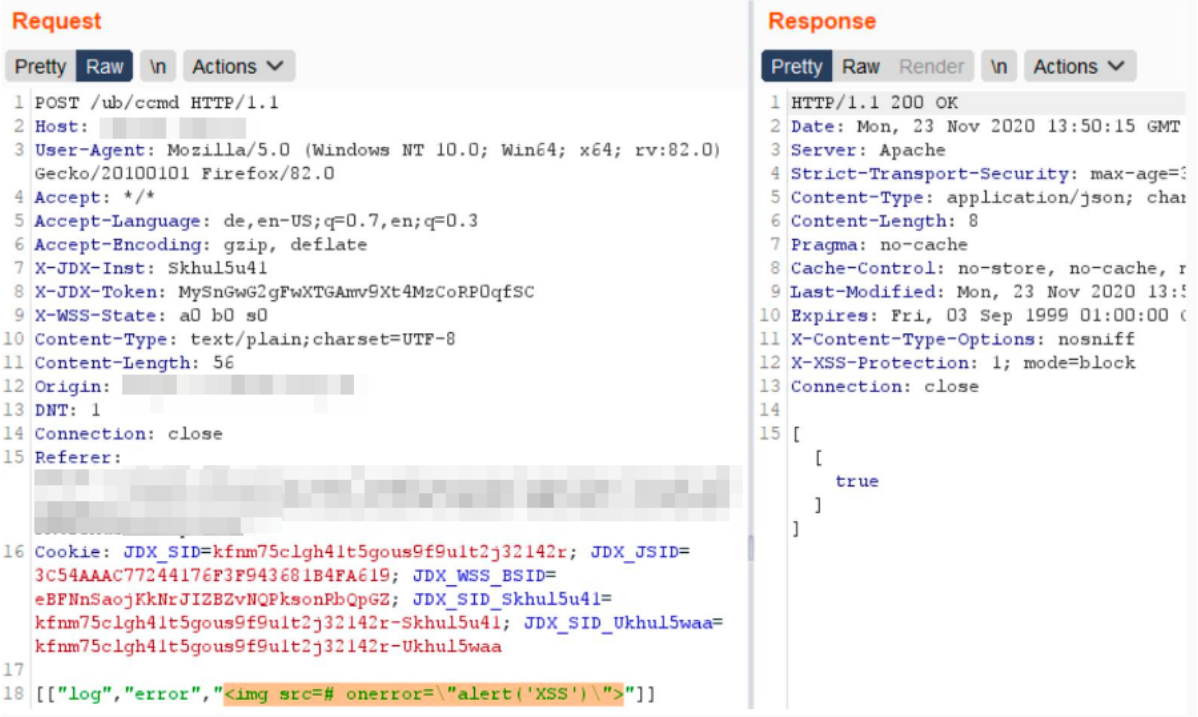

Figure 16 - Persistent Cross-Site Scripting: Proof of Concept Request

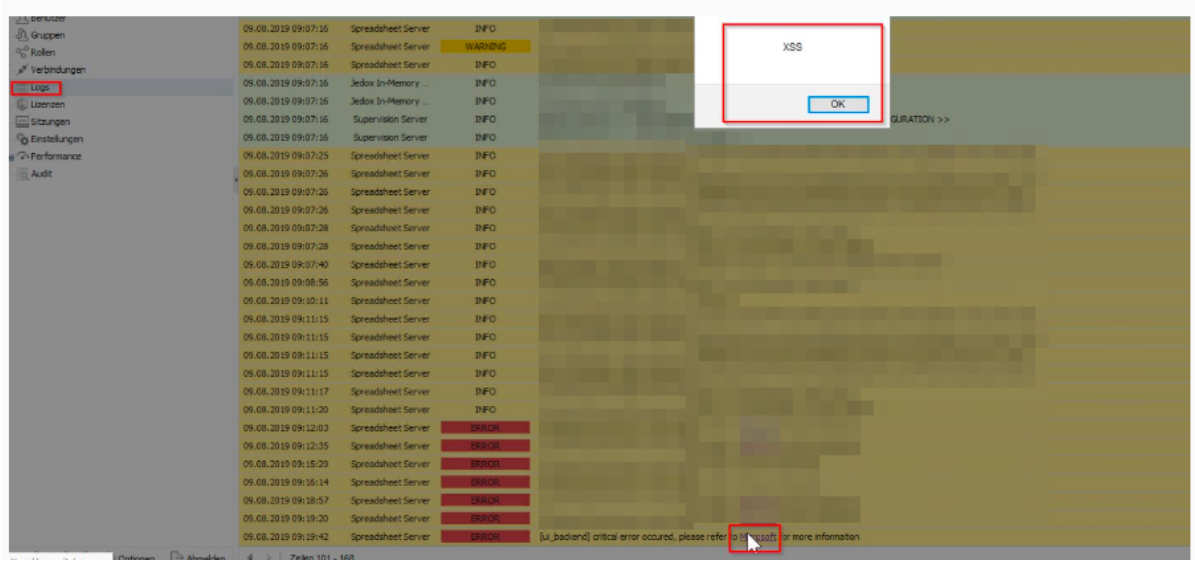

Figure 17 - Execution of the XSS payload

The injected JavaScript code is executed in an administrator's log view and could have allowed an attacker to elevate privileges.

CVE-2022-47876: Remote Code Execution via Executable Groovy-Scripts

Criticality: Critical
CVSS-Score: 9.1
Vulnerability Type: Exec Code
Product: Jedox 2020.2.5
Affects: Component: Integrator

Overview

Integrator in Jedox 2020.2.5 allows remote authenticated users to create Jobs to execute arbitrary code via Groovy-scripts. To exploit the vulnerability, the attacker must be able to create a Groovy-Job in Integrator.

Description

A user with appropriate permissions can create Groovy jobs in the Integrator with custom source code and then execute them.

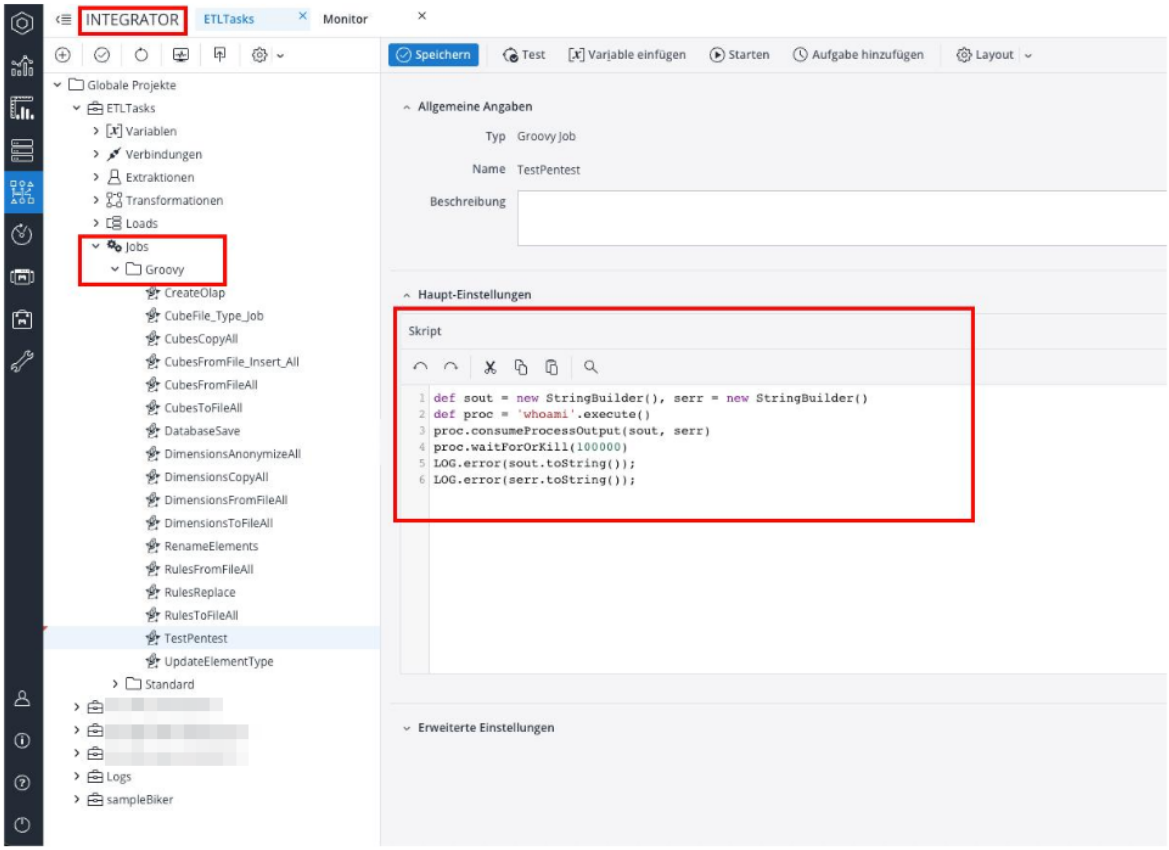

Figure 18 - Creating the Groovy Job

Among other things, the following code can be used to execute arbitrary commands on the system:

```
def sout = new StringBuilder(), serr = new StringBuilder()
def proc = 'whoami'.execute()
proc.consumeProcessOutput(sout, serr)
proc.waitForOrKill(10000)
LOG.error(sout.toString());
LOG.error(serr.toString());
```

CVE-2022-47878: Remote Code Execution via Configurable Storage Path

Criticality: Critical
CVSS-Score: 9.1
Vulnerability Type: Incorrect Input Validation
Product: Jedox 2020.2.5
Affects: Setting: default-storage-path

Overview

Incorrect input validation for the default-storage-path in the settings page in Jedox 2020.2.5 allows remote, authenticated users to specify the location as web root directory. Consecutive file uploads can lead to the execution of arbitrary code. To exploit the vulnerability, the attacker sets the default storage path to the web root.

Description

In the application settings the default storage path can be set to any value.

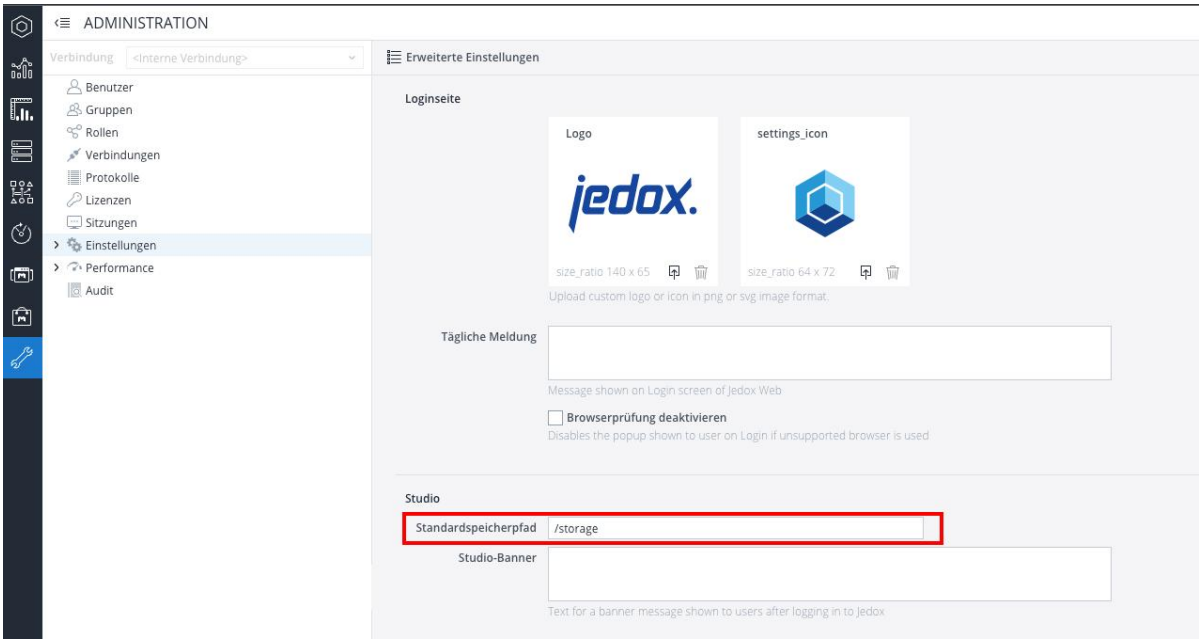

Figure 19 - Default storage path setting

An attacker can now set this path to, for example, a directory on the web server such as `/htdocs/app/docroot/be/`.

Then, the upload/import function can be used to upload a malicious .php file to a new root directory.

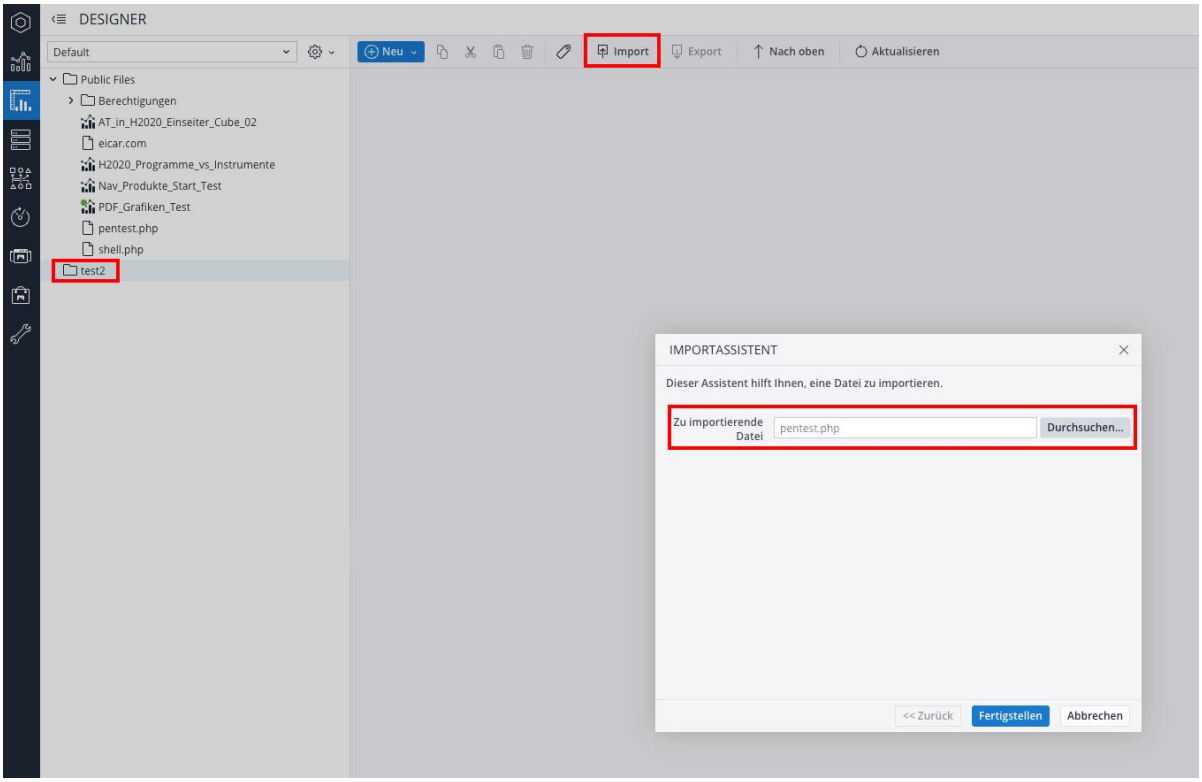

Figure 20 - Importing the file "pentest.php"

As seen in the server response of the file upload, the file ends up in the directory specified by the attacker.

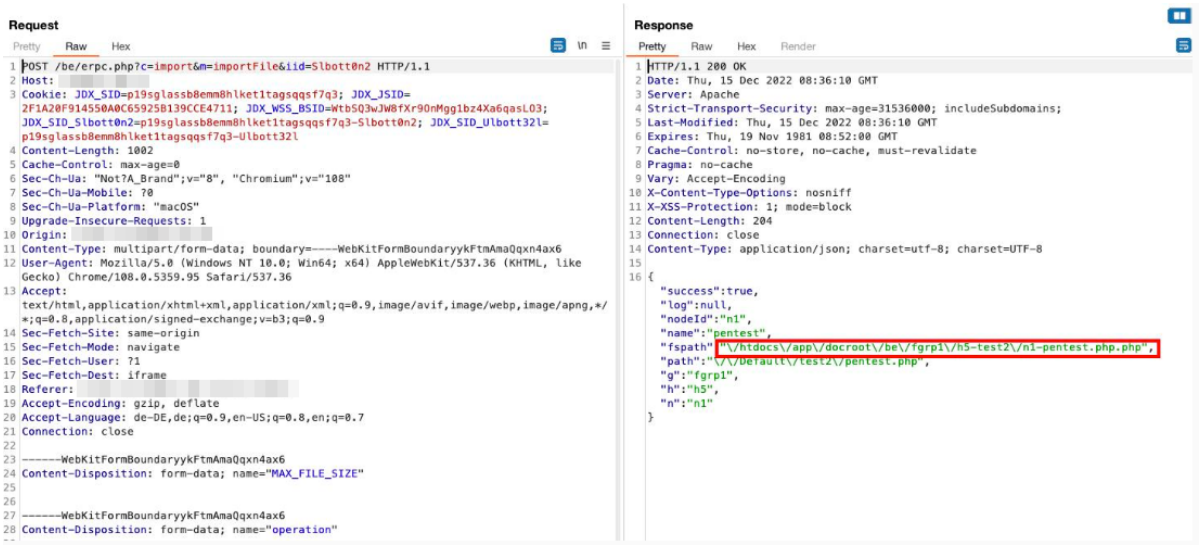

Figure 21 - Location of the file

The uploaded file can now be executed directly from the web server:

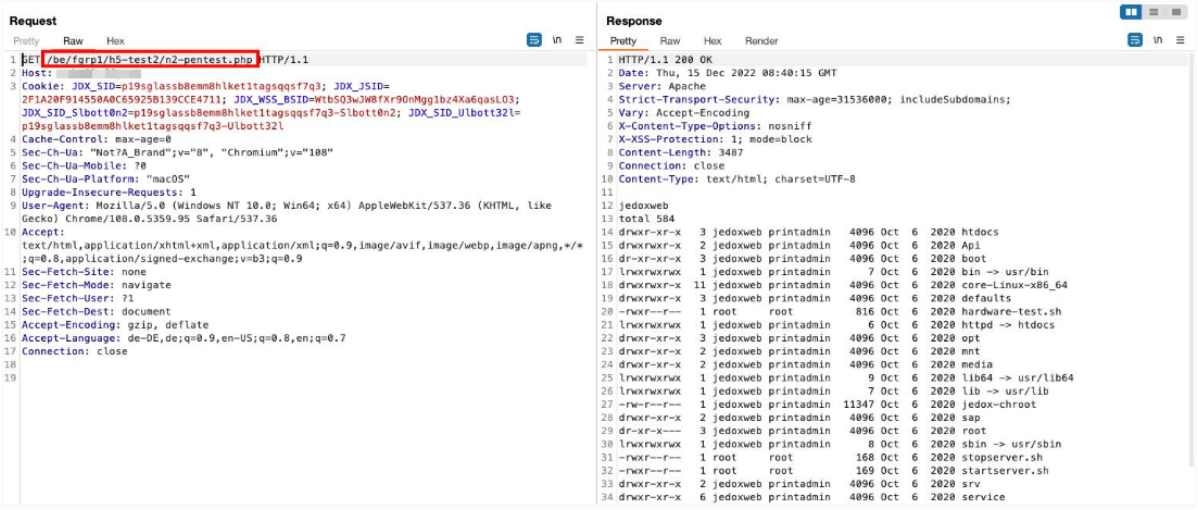

Figure 22 - Execute the uploaded file

CVE-2022-47874: Disclosure of Database Credentials via Improper Access Controls

Criticality: High
CVSS-Score: 7.7
Vulnerability Type: Incorrect Access Control
Product:
- Jedox Cloud
- Jedox 2020.2.5

Affects:
- Component: /tc/rpc
- Component: com.jedox.etl.mngr.Connection::getGlobalConnection

Overview

Improper access controls in */tc/rpc* in Jedox Cloud and Jedox 2020.2.5 allows remote authenticated users to view details of database connections via the class *com.jedox.etl.mngr.Connections* and the method *getGlobalConnection*. To exploit the vulnerability, the attacker must know the name of the database connection.

Description

A list of all database connections can be retrieved via `conn::ls`:

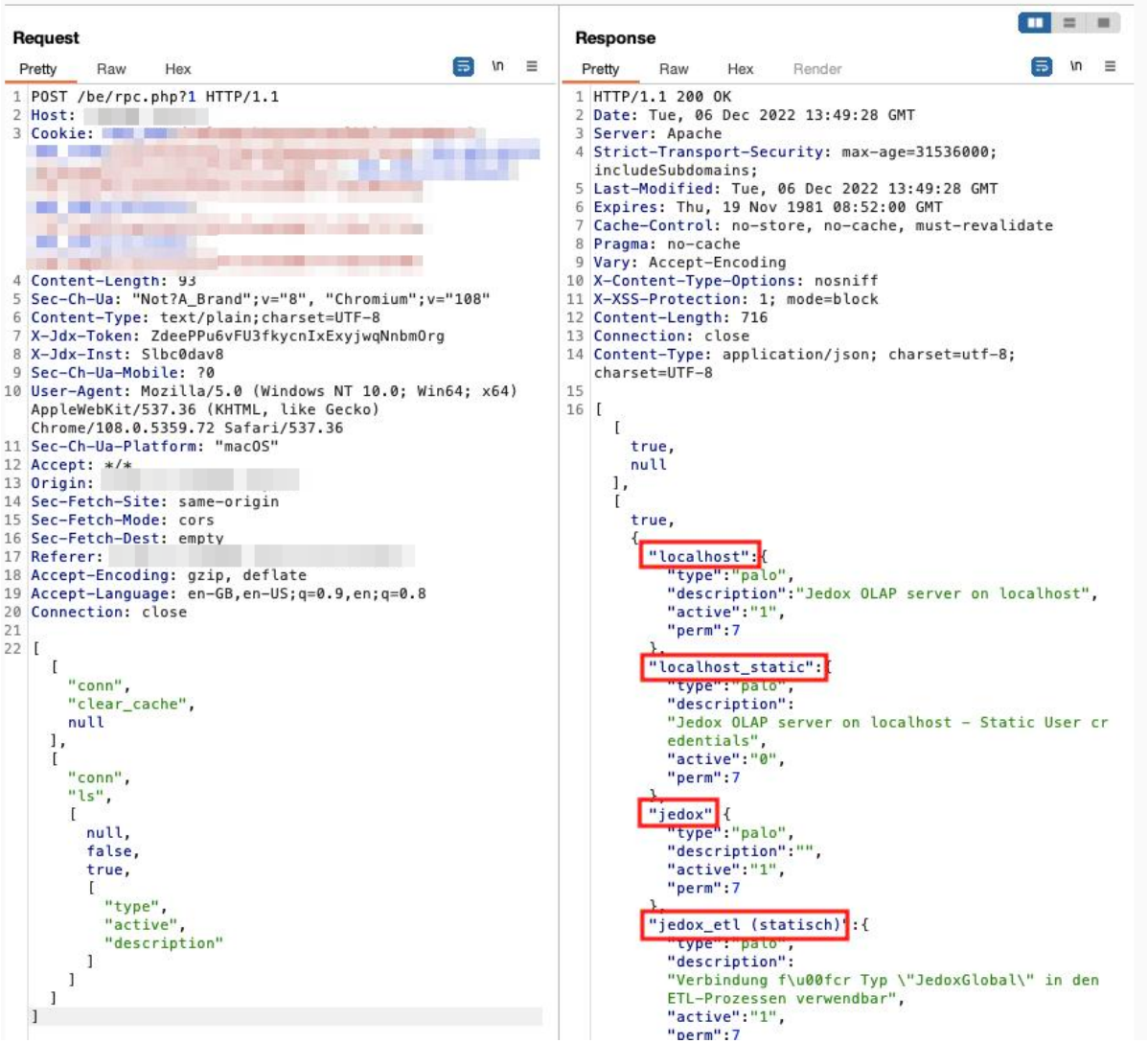

Figure 23 - Query the list of all db connections

Details of individual database connections (including encrypted credentials) can be retrieved using the Java RPC function `com.jedox.etl.mngr.Connection::getGlobalConnection`.

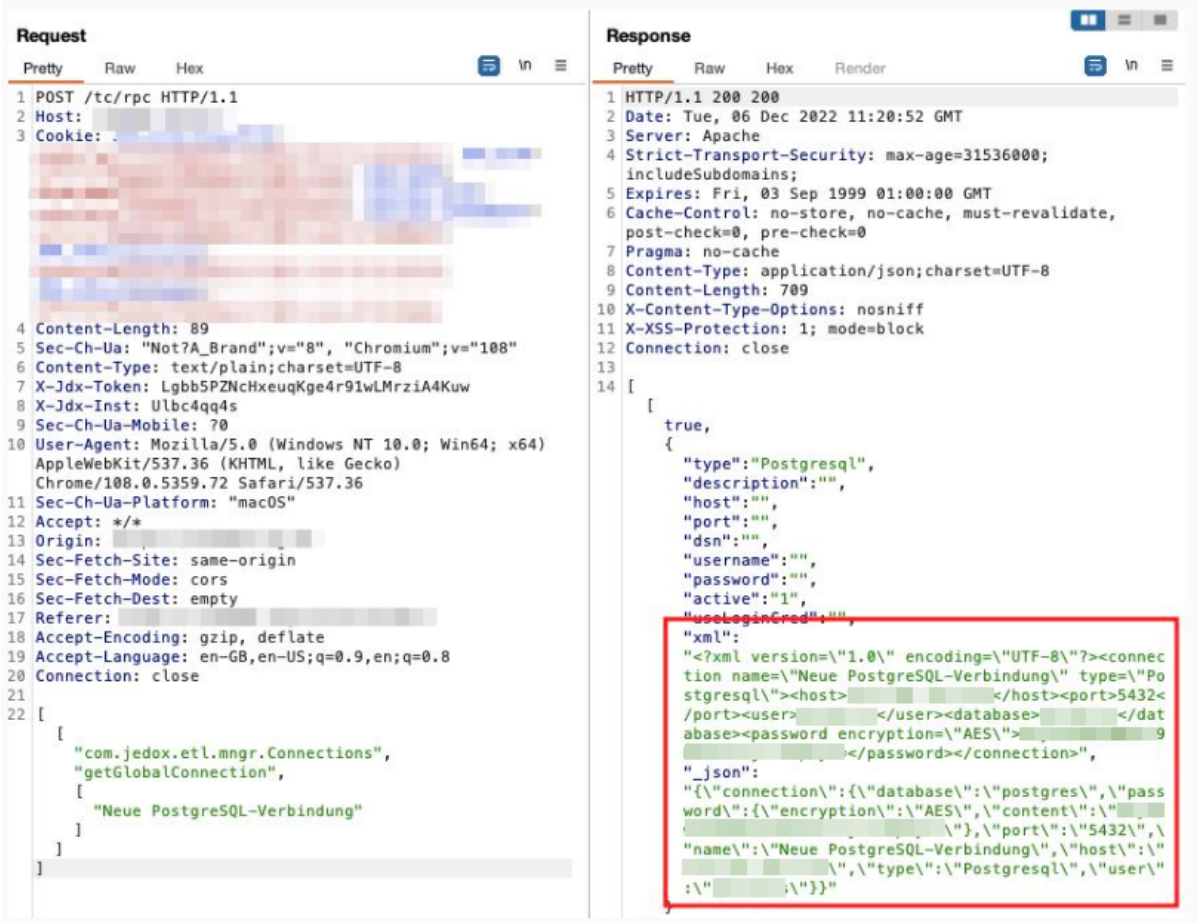

Figure 24 - Retrieve connection details

The credentials requested via this method are encrypted, however they can be decrypted via CVE-2022-47879.

CVE-2022-47880: Disclosure of Database Credentials via Connection Checks

Criticality: Medium
CVSS-Score: 6.8
Vulnerability Type: Information Disclosure
Product:
- Jedox Cloud
- Jedox 2020.2.5

Affects:
- Settings Page
- Component: /be/rpc.php

Overview

An information disclosure vulnerability in `/be/rpc.php` in Jedox Cloud and Jedox 2020.2.5 allows remote authenticated users with the appropriate permissions to modify database connections to disclose the clear text credentials via the `test connection` function. To exploit the vulnerability, the attacker must set the host of the database connection to a server under his control.

Description

The host part of a database connection can be changed without the need to re-authenticate. Afterwards the connection can be tested:

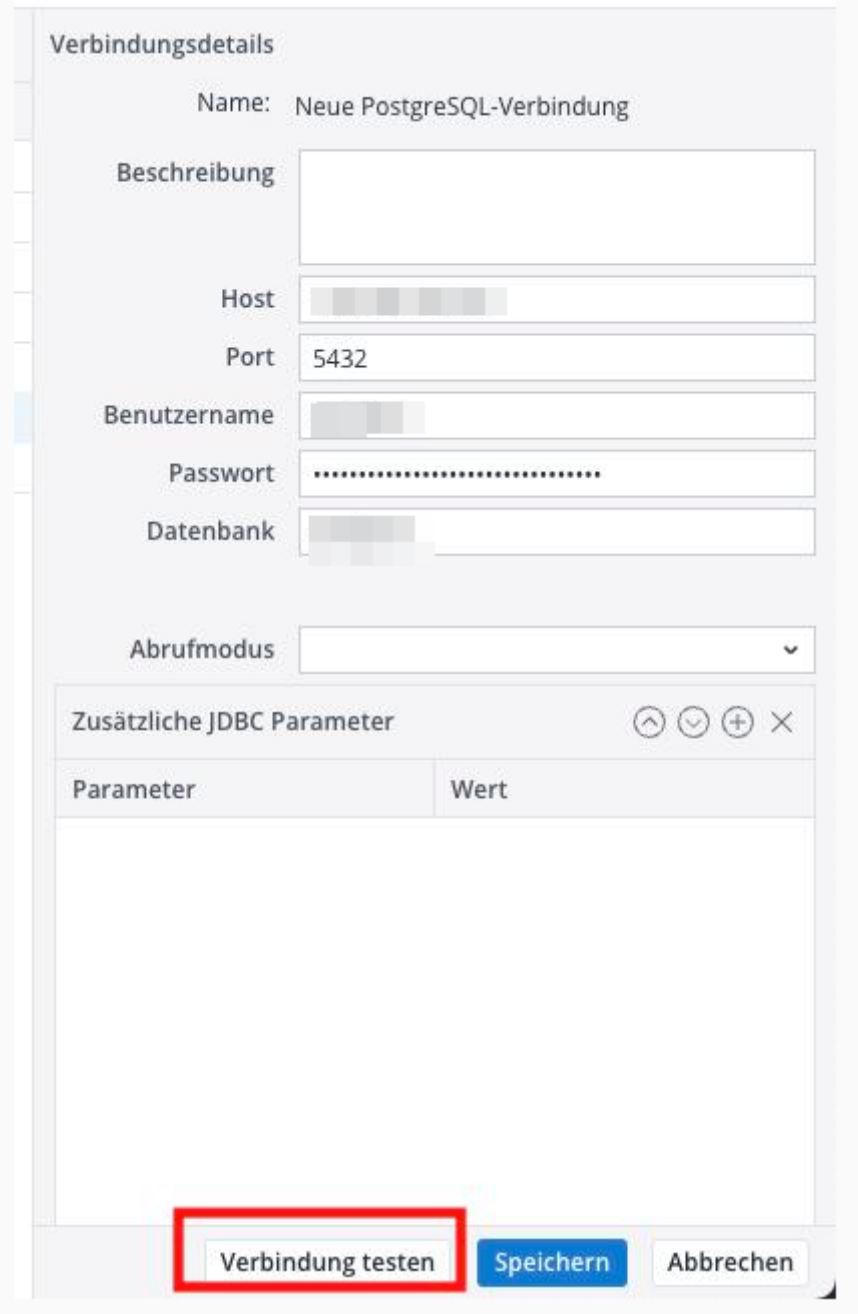

Figure 25 - Change host and test connection

This sends the clear text credentials, which are not actually known to the user, to the server controlled by the attacker:

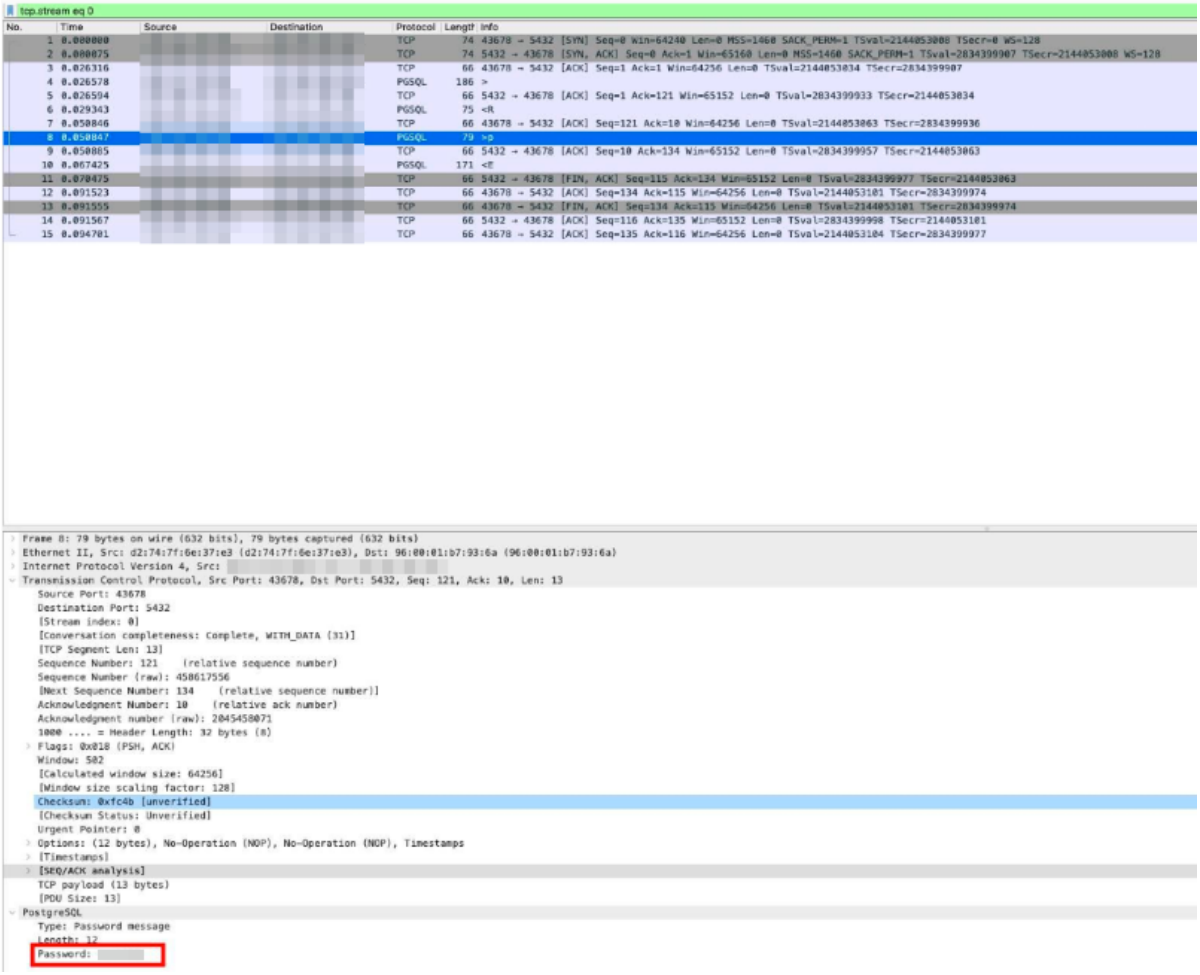

Figure 26 - Clear text credentials in Wireshark

Timeline

Date	Description
2022-12-20	Initial contact to the vendor via two managers
2022-12-27	Contact with vendor via public mail address
2023-01-11	Vendor provides encrypted channel for vulnerbility information
2023-01-18	Reporting of vulnerability details
2023-04-28	Planned public disclosure

Imprint

syslifters.com | **Dedicated to Pentests.**
Syslifters GmbH | Eitzersthal 75 | 2013 Göllersdorf
FN 578505 v | District Court Hollabrunn

End of Report

This report was rendered
by SysReptor with
♥