Technical Impact Assessment

ForgeRock AM CVE-2021-35464

This document is intended to support security teams in identification of CVE-2021-35464, along with success and failure of exploitation. This document also gives basic examples of testing, detections, understanding simple payloads, and techniques to reverse them to support investigations.

Impacts of exploitation

The RCE impact of CVE-2021-35464 shares most common traits of RCE within Java deserialization attacks. An attacker exploiting the vulnerability will execute commands in the context of the current user, not as the root user (unless ForgeRock AM is running as the root user, which is not recommended). Thus will be limited to the scope of the user.

An attacker can use the code execution to extract credentials and certificates, or to gain a further foothold on the host by staging some kind of shell (such as the common implant Cobalt Strike).

Ensure your ForgeRock AM is running as a user with minimal privileges. The ForgeRock AM deployment should also have suitable firewall, or security groups attached which do not allow traffic in or out of the server where not explicitly required.

Testing for Exploitability

There are two methods of testing using BurpSuite, the other can create logs/files on the server hosting AM. For the BurpSuite root please see the PortSwigger blog

Please be cautious when testing against production servers.

Creating tmp files

If you have access to the server a simple way to check which does not require timing or callback requests is to create a payload which creates a file.

You will need to download ysoserial[1].

To create the base64 payload run the following command, if you are using Windows to deploy ForgeRock AM you will need to alter the touch command.

```
java -jar ysoserial-master-d367e379d9-1.jar Click1 "touch
/tmp/cve-20201-35464_text.txt" | (echo -ne \\x00 && cat) | base64
| tr '/+' '_-' | tr -d '='
```

The payload can be delivered by POST, or GET request. Examples of which can be seen in the Detecting Exploitation section.

Detecting Exploitation

In order to exploit the vulnerability threat actors must be able to craft a HTTP request to one of the endpoints listed below:

- `/ccversion/Version`
- `/ccversion/Masthead`
- `/ccversion/ButtonFrame`

At the time of writing the list of known endpoints, but as a precaution consider any endpoints containing "`/ccversion/*`" to be at risk of exploitation.

The payload can be successfully sent via GET or POST requests using the `jato.pageSession` containing a base64 encoded payload.

The HTTP method used to deliver the payload has been identified in the wild using both GET and POST methods. However, it may be possible for threat actors to use a variety of HTTP methods, thus it is important that SoC and DFIR teams do not narrow down on these specific methods.

Using the fix provided in the advisory unsuccessful exploitation attempts will return a 404, this should support SoC teams from

HTTP Status Code

In the event a threat actor attempts to exploit the vulnerability the server will likely return a 302 status code if the exploitation of an unpatch server occurs. This can be a strong indicator that the payload was delivered and executed. It is important to note that the state and success of any second stage payload is unknown.

Initial Patch
If an organisation uses the initial patch released on June 29th, which requires the edit of the web.xml file, any exploit attempts will return a 404 status code.

Official Fix
If you have deployed the official ForgeRock patch when a threat actor attempts the exploit the vulnerability they will receive a 302 redirecting them to the following URL:
`https://yourdomain.com/openam/base/AMUncaughtException`

Example GET request

```
GET /openam/oauth2/..;/ccversion/Version?jato.pageSession=AKztAAVzcgA[...]XamF2YS5
HTTP/1.1
Host: authentication.example.com
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:81.0) Gecko/20100101
Firefox/81.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
```

```
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
```

Example POST request

```
POST /openam/ccversion/Masthead HTTP/1.1
Host: authentication.example.com
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:81.0) Gecko/20100101
Firefox/81.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Connection: close
Upgrade-Insecure-Requests: 1
Content-Length: [...]


jato.pageSession=AKztAAVzcgA[...]XamF2YS5
```

Understanding the payload

Due to the early stages of the PoC being public there are few variants of the exploit, but it is important for the SoC team to continue to review attacks for mutation which may bypass alerting.

Depending on the attacker using the “Click1” ysoserial gadget the payload when decoded should have a single reference to “exec”, on the same line before this is where the attacker embedded command is.

See the example from our previous payload:

It should be noted that more complex payloads may not be as trivial to decode and may take more effort to reverse to an understandable state.

Process Creation

For organisations with EDR in place it may be possible to detect malicious activity on the server running ForgeRock AM. Under normal and default operating conditions the ForgeRock AM process does not spawn children processes (after the initial configuration), so detecting child processes may indicate malicious activity.

There are some caveats to this indicator:

- ForgeRock AM installations running on Windows Servers require a subprocess during authentication which may create false positives.
- Customers using custom authentication scripts might result in child processes being spawned.
- Often the child processes are very short lived, and can be easily missed if using tools such as `ps` periodically.

This is a non-exhaustive list of caveats and we will update where possible.

References

1. https://github.com/frohoff/ysoserial

Document Control

Version	Author Dept	Date	Comments
Initial Draft	Enterprise Security	2nd July 2021	-
Final Draft	Enterprise Security	2nd July 2021	Ready for BackStage release.
Update	Enterprise Security	13th July	New section: Process Creation, moved links to ref section.
Update	Enterprise Security	15th July	Updated information on detecting exploitation, endpoints, and patching results.