Action required: Atlassian Connect vulnerability allows bypass of app qsh verification via context JWTs

jira-cloud, confluence-cloud, atlassian-connect, security, jwt

HeyJoe Atlassian Staff Apr '21

- [Update May 7 2021] - Please upgrade to Atlassian Connect Spring Boot 2.1.5 or later, 2.1.4 regressed on the fix introduced in 2.1.3

What is changing?

Atlassian is making a breaking API change to Atlassian Connect in Jira and Confluence Cloud to mitigate a vulnerability in the verification of the qsh claim in Connect JWT authentication.

Is my app impacted?

If your app uses an **authentication descriptor setting** **other than** none, then it's highly likely your app is impacted and will need to be updated.

This vulnerability affects apps regardless of how they are installed; apps listed on the Marketplace and apps installed via the UPM "dev mode" **are affected** .

If your app does not use **context JWTs** (see below), you **must still opt-in** to the breaking changes as soon as possible.

Apps running on Bitbucket Connect are **not affected** .

Why is it changing?

When Atlassian makes a request from Jira or Confluence to a Connect app there's two forms of authentication that are used depending on how the app is embedded in the products:

1. **An iframe or server-to-server JWT** : For **Connect modules** (web pages, web panels, etc) and lifecycle hooks
2. **A context JWT** : For securely passing product UI context to an app's server as described in the **cacheable app iframes guide** .

Connect apps are expected to validate these tokens to verify the authenticity of the request. The verified JWT will tell the app which user is interacting with the app, which part of the app they're interacting with, and the context for where the app is embedded inside Jira or Confluence.

Because **iframe JWTs** are exposed to end users, they contain a qsh (query string claim) that provides a way for Connect apps to validate that the app URL and query string haven't been tampered with. **Context JWTs** do not contain a qsh claim, and are free to be used against any app endpoint. This is problematic as:

1. The **Connect JWT specification** does not clearly document how apps should handle **iframe JWTs** and **context JWTs** differently (as the difference between them is subtle).
2. Some apps, and app frameworks, may accept **context JWTs** in the place of **iframe JWTs** which may bypass the URL tampering checks described above.

What do I need to do?

We advise vendors to follow the **app incident management guidelines** if you determine exploitation of the vulnerability within your environment.

To prevent **context JWTs** from being accepted by default for any app endpoint, Atlassian is planning to add qsh claims to **context JWTs** for all apps by Jun 7, 2021. As soon as possible, we ask that you that follow the instructions below to ensure your app is compliant with these changes.

Running an Atlassian-supported Connect framework

1. Update **Atlassian Connect Spring Boot** , or **Atlassian Connect Express** to the latest version; and

- [Update May 7 2021] - Please upgrade to Atlassian Connect Spring Boot 2.1.5 or later, 2.1.4 regressed on the fix introduced in 2.1.3

2. Opt-in to the updated **context JWTs** by adding a new context-qsh element to the apiMigrations section of the app descriptor.†
3. Update your app endpoints that are designed to work with **context JWTs** to accept a static qsh of context-qsh‡
4. We **strongly recommend** having separate endpoints for each type of JWT. To avoid subverting qsh iframe validation you should not accept both **context JWTs** and **iframe JWTs** on a single endpoint.
5. The qsh enforcement in these new framework versions is backwards compatible, so the change will work with both tenants that have the updated qsh **context JWTs** and tenants that are yet to upgrade your app
6. Test and deploy your app.

† Your descriptor should look like this:

```
...
  "apiMigrations": {
     "context-qsh": true
```

Skip to main content

```
  }
...
```

‡ for Atlassian Connect Express set skipQshVerification by passing true to addon.authenticate. This will allow the endpoint to be called with a **context JWT** .

```
app.get('/example', addon.authenticate(true), (req, res) => {
  // ...
  res.status(200).send('ok')
});
```

If your app already uses addon.authenticate() for your **iframe or server-to-server** endpoints and addon.authenticate(true) (or addon.checkValidToken()) for the endpoints your front-end uses, then you do not need to change your Atlassian Connect Express app code.

For Atlassian Connect Spring Boot annotate methods with @ContextJWT:

```
@ContextJwt
@RequestMapping(value = "/example", method = RequestMethod.GET)
public ResponseEntity<Void> exampleEndpoint(@AuthenticationPrincipal AtlassianH
  // ...
  return ResponseEntity.ok().build();
}
```

Running a custom implementation

If you're using a framework other than Atlassian Connect Spring Boot, or Atlassian Connect Express, then please follow these steps to ensure your app correctly verifies the qsh claim:

1. Opt-in to the updated **context JWTs** by adding a new context-qsh element to the apiMigrations section of the app descriptor.†
2. Follow **Understanding JWT for Connect apps** to validate JWTs using your app secret
3. Reject any JWT that does not have a qsh claim
4. For app endpoints that need to accept a context JWT, such as requests from your app front-end to back-end:

- Validate the qsh claim is set to context-qsh instead of building a query string hash from the request path and parameters
- Do not apply this to all endpoints by default, only for endpoints that need to accept context JWTs

† Your descriptor should look like this:

```
...
  "apiMigrations": {
     "context-qsh": true
  }
...
```

For reference, a pseudo-code example of an endpoint that accepts context JWTs:

```
Function HandleAPIRequest(request)
  JWT <- request.Header['Authorization'] or request.QueryString['jwt']

  claims <- ValidateJWTSignature(JWT)

  if not claims then
    // JWT signature validation failed
    return Error(Unauthorized)

  if not claims.qsh then
    // Missing QSH claim - reject
    return Error(Unauthorized)

  // Normal QSH verification based on the request parameters
  requestQsh <- ComputeQSH(request)
  if requestQsh != claims.qsh and claims.qsh != 'context-qsh' then
    // QSH verification failed
    return Error(Unauthorized)

  // success, let the request through
  return OK
```

By when do I need to do it?

Under the recently announced **Vulnerability Management Program for Marketplace Apps**, Atlassian will create AMS issues for eligible apps with a target SLA to patch this vulnerability.

To reduce the security impact of this vulnerability, the change will be automatically applied to all apps that haven't used context JWTs for the last two weeks. There should be no impact as a result of this change. If your app has been incorrectly identified as being unaffected, please create an issue in our **developer support portal** for this change to be reversed for your app.

Enforcement of this breaking change is planned for all apps byJun 7, 2021. If you do not patch your app to mitigate this vulnerability and opt-in to the context-qsh API migration before this date, your app may stop working.

Please reply to this announcement if you have any questions about this change, or if you need to discuss the situation in more detail.

Context jwt vulnarbility

How to use ACE addon.authenticate(true)

Send a request from an iframe is receiving a 401

Error 401 in our JIRA App while we're actually authenticated

Registering a Jira add on

6 more

Alex4 **Apr '21**

Hi, thanks for the instructions!
Please, Could you also update the code examples in the repository? These examples of the powerful help!

- **atlassian-connect-spring-boot-sample-dynamodb(jwt + db)**
- **atlassian-connect-spring-boot-samples**

---

colin.hammond **Apr '21**

I am struggling to follow this documentation, yet our business depends on handling this change correctly.

Can we please have some practical examples of what needs to be done with an Atlassian-Connect app please.

Is there a tool that is available that can examine our Atlassian-connect doc and api and give specifics about what needs to be done to make it compliant, ie the delta?

---

remie Marketplace Partner **Apr '21**

Here is my take at making this more digestible:

For all Connect apps

Adjust descriptor of all your connect apps
Regardless of whether you need to actually make any changes to the logic of your apps, you will need to add the following to your descriptor:

```
"apiMigrations": {
     "context-qsh": true
  }
```

This is mandatory for all Connect apps to be implemented before June 7th, 2021. Failure to do so will result in a vulnerability report on the Security dashboard with an SLA of 4 weeks to comply. Failure to mitigate the vulnerability might result in delisting of your app.

For Connect apps running on Atlassian Connect Express (ACE)

Endpoints that serve iframe content, webhooks or lifecycle events
Atlassian will add a valid QSH to all endpoints that serve server-side rendered HTML content for iframes, webhooks or app lifecycle events (except for the first installation). Basically any URL that is listed in your descriptor. Those endpoints will need to enforce QSH validation. For these endpoints, you probably do not need to change anything, as you will already have `addon.authenticate()` in the middleware, like this:

```
app.get('/path/to/webitem', addon.authenticate(), (req, res) => {
  // return server-side rendered HTML
});
```

API endpoints that are called from front-end
Calls to these endpoints do not have valid QSH claims because the JWT token is generated on the front-end without any context using `AP.context.getToken()`.

For these endpoints, which you can identify in your front-end code, you will need to disable JWT verification by changing the code from `addon.authenticate()` to `addon.authenticate(true)`:

```
app.get('/path/to/api/endpoint', addon.authenticate(true), (req, res) => {
  // will probably return some JSON
});
```

For Connect apps running on Atlassian Spring Boot
Update to the latest version of Atlassian Spring Boot.

Endpoints that serve iframe content, webhooks or lifecycle events
Atlassian will add a valid QSH to all endpoints that serve server-side rendered HTML content for iframes, webhooks or app lifecycle events (except for the first installation). Basically any URL that is listed in your descriptor. Those endpoints will need to enforce QSH validation. For these endpoints, you do not need to change anything.

API endpoints that are called from front-end
Calls to these endpoints do not have valid QSH claims because the JWT token is generated on the front-end without any context using `AP.context.getToken()`.

For these endpoints, which you can identify in your front-end code, you will need to add `:extJWT` annotation:

Skip to main content

```
@ContextJwt
@RequestMapping(value = "/path/to/api/endpoint", method = RequestMethod.GET)
public ResponseEntity<Void> exampleEndpoint(@AuthenticationPrincipal AtlassianH
  // ...
  return ResponseEntity.ok().build();
}
```

For Connect apps running on custom implementation

Endpoints that serve iframe content, webhooks or lifecycle events
Atlassian will add a valid QSH to all endpoints that serve server-side rendered HTML content for iframes, webhooks or app lifecycle events (except for the first installation). Basically any URL that is listed in your descriptor. Those endpoints will need to enforce QSH validation. For these endpoints, you probably do not need to change anything as you are already doing it (or decided not to).

API endpoints that are called from front-end
Calls to these endpoints do not have valid QSH claims because the JWT token is generated on the front-end without any context using `AP.context.getToken()`.

For these endpoints, which you can identify in your front-end code, you will need to add QSH validation and enforce it by ensuring the `qsh` claim is part of the JWT and is set to value `context-jwt`.

401 Unauthorized when using jwt token in my plugin

After updated atlassian-connect-spring-boot-starter from 1.3.2 to 2.1.5 then getting erro...

---

marc Marketplace Partner **Apr '21**

Hi @colin.hammond Maybe it helps, I've published a blog post which tries to explain the steps and a bit of context: **Atlassian Connect Security – April 2021 changes – H-RD.ORG**

---

nathanwaters Marketplace Partner **Apr '21**

For what appears to be a mandatory marketplace-wide change, these instructions are incredibly obtuse. If my apps use "none" for the authentication descriptor do I need to do anything?

---

cmacneill **Atlassian Staff** **Apr '21**

Apps that use "none" for their authentication settings do not need to take any action.

---

dortiz **Apr '21**

Hi @cmacneill ,

as I understand it, if my app use **"none"** for the **authentication descriptor**, it is **only** necessary to add **"context-qsh": true** in the **"apiMigrations" section** to avoid generating the vulnerability report. it is right?

Thank you

---

colin.hammond **Apr '21**

Thank you Remie,

This is very helpful indeed. I really appreciate you helping me and others by explaining this in more detail.

It's so important to get this right and potentially business damaging if we don't.

---

colin.hammond **Apr '21**

Thanks marc. All background and guidance is helpful.

---

zsims **Atlassian Staff** **Apr '21**

> dortiz:
>
> if my app use **"none"** for the **authentication descriptor** , it is **only** necessary to add **"context-qsh": true** in the **"apiMigrations" section** to avoid generating the vulnerability report. it is right?

Your understanding is correct. Your app isn't vulnerable if your app uses `none` for `authentication`.

We'll only be raising **AMS tickets** for eligible apps that haven't opted into the `apiMigrations` and are using authentication **other than** none.

---

dortiz **Apr '21**

Skip to main content

Thank you very much for the reply

JulienJulien **Apr '21**

Thanks for the explanations.

Some of the HTML links describes in the descriptor, such as dialogs, cannot be validated (JWT/QSH) because JWT token is not sent with the request: **AP.Dialog doesn't provide JWT token on URL call**

Is it planned to also include these for JWT/QSH validation and so, include JWT/QSH in the request ?

Thanks for your reply.

---

dciupureanu Marketplace Partner **Apr '21**

Hi,

Thanks for the write-up about the vulnerability and the fix. Since this is a security vulnerability, do you have a CVSS score for it?

Thanks.

OfirNir Cloud Developer **Apr '21**

NVD - CVE-2021-26074 - Connect Spring Boot
NVD - CVE-2021-26073 - Connect Express

---

zsims **Atlassian Staff** **Apr '21**

> dciupureanu:
>
> Since this is a security vulnerability, do you have a CVSS score for it?

Great question. As @OfirNir linked, we've raised CVEs (currently undergoing analysis, will be scored in the NVD soon) for Atlassian Connect Express and Atlassian Connect Spring Boot (the official Atlassian Connect app frameworks). These vulnerabilities cover the issue of context JWTs being accepted in lifecycle endpoints:

- **CVE-2021-26074** - Atlassian Connect Spring Boot - **CVSS v3 9.1 - Critical**
- **CVE-2021-26073** - ACE - **CVSS v3 9.1 - Critical**
- [Update May 11 2021] **CVE-2021-26077** - Atlassian Connect Spring Boot 2.1.4 regression - **CVSS v3 9.1 - Critical**

The overall vulnerability, which is due to ambiguity of context JWTs and iframe/server-to-server JWTs, was scored also as **CVSS v3 9.1 Critical** which was upgraded from the initial `CVSS High` scoring after additional analysis.

---

paulo.alves Marketplace Partner **Apr '21**

Hi,

Does this means that all the vendors should threat this as a security incident as mentioned here? **https://developer.atlassian.com/platform/marketplace/app-security-incident-management-guidelines/**

Or is Atlassian planning to make a single communication to all customers that have apps installed?
I'm not sure about the model that Atlassian uses to report vulnerabilities on cloud products. I don't remember to be notified about a critical vulnerability on jira, confluence or bitbucket cloud in the last 3 years. Is it possible to check any advisory, or Atlassian never had a critical vulnerability reported for cloud products?

Thank you.

---

remie Marketplace Partner **Apr '21**

> paulo.alves:
>
> Does this means that all the vendors should threat this as a security incident [...]

Well, either way, not all vendors. Only those using ACE / ACSB. We're not affected by this vulnerability, because we have a custom implementation and never accepted context JWT for lifecycle events, webhooks, etc.

---

paulo.alves Marketplace Partner **Apr '21**

Sure, I was referring to all vendors using ACE or ACSB.

---

zsims **Atlassian Staff** **Apr '21**

> paulo.alves:

Skip to main content

> Does this means that all the vendors should threat this as a security incident as mentioned here?
> **https://developer.atlassian.com/platform/marketplace/app-security-incident-management-guidelines/**

We advise vendors to follow the **app incident management guidelines** if you determine exploitation of the vulnerability within your environment.

> paulo.alves:
>
> Or is Atlassian planning to make a single communication to all customers that have apps installed?
> I'm not sure about the model that Atlassian uses to report vulnerabilities on cloud products. I don't remember to be notified about a critical vulnerability on jira, confluence or bitbucket cloud in the last 3 years. Is it possible to check any advisory, or Atlassian never had a critical vulnerability reported for cloud products?

Good question, we contact customers inline with the Atlassian Security Practices. You can read more about them on our Trust Center: **Security Practices | Atlassian**
Server advisories (which require customer action) are posted on **Security Advisories | Atlassian**