

**Finding Name:** Improper Verification of Cryptographic Signature

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Team** | **Role** | **Project** | **Quality Assurance** | **Is this a re-tested Finding?** |
| Payas Paul | SCR | Senior Lead | Ontrack |  |  |
|  |  |  |  |  |  |

|  |
| --- |
| **Was this Finding Successful?** |
| Yes |

**Finding Description**

The json-jwt Ruby gem's vulnerability stems from incorrect cryptographic signature verification. The issue specifically results from the token verification process's confusion between the encryption and signature techniques.   
  
When JSON::JWT.decode is used to decode JSON Web Tokens (JWTs), the library can handle the cryptographic signature verification incorrectly. This misconception can be used by an attacker to get around identity checks, which could result in unwanted access or other security problems.

**Risk Rating**  
Impact: Significant   
Likelihood: Moderate

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Impact values** | | | | |
| **Very Minor** | **Minor** | **Significant** | **Major** | **Severe** |
| Risk that holds little to no impact. Will not cause damage and regular activity can continue. | Risk that holds minor form of impact, but not significant enough to be of threat. Can cause some damage but not enough to impede regular activity. | Risk that holds enough impact to be somewhat of a threat. Will cause damage that can impede regular activity but will be able to run normally. | Risk that holds major impact to be of threat. Will cause damage that will impede regular activity and will not be able to run normally. | Risk that holds severe impact and is a threat. Will cause critical damage that can cease activity to be run. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Likelihood** | | | | |
| **Rare** | **Unlikely** | **Moderate** | **High** | **Certain** |
| Event may occur and/or if it did, it happens in specific circumstances. | Event could occur occasionally and/or could happen (at some point) | Event may occur and/or happens. | Event occurs at times and/or probably happens a lot. | Event is occurring now and/or happens frequently. |

**Business Impact**

The following are some possible outcomes:   
  
Security Vulnerability: By creating a malicious JWT that the program would erroneously validate, an attacker might take advantage of this vulnerability and gain unauthorized access to private information or program functionality.   
  
Data Integrity: Should the application depend on JWTs to guarantee data integrity, an inaccurate JWT verification could lead to undetected data modification or tampering.   
  
Loss of Trust: If the vulnerability is used to gain an advantage, consumers and clients may become less trusting of you, particularly if confidential data is exposed.

Legal and Compliance Issues: Depending on the type of data impacted by the vulnerability, the business may be vulnerable to legal action as well as compliance problems, particularly if the data is governed by laws like GDPR or HIPAA.   
  
Financial Loss: The organization may sustain financial losses as a result of remediation costs, legal bills, and possible fines brought on by a security breach.

**Affected Assets**

This vulnerability affects any system or data that depends on JSON Web Tokens (JWTs) being correctly verified in order to authenticate or authorize users. This might consist of:   
  
User Data: Should the vulnerability be used to obtain unauthorized access, any user data that the application stores or processes could be impacted.   
  
Sensitive Data: Sensitive data, such financial information or personally identifiable information (PII), may be in danger if the application employs JWTs to secure it.   
  
Application Functionality: An attacker may be able to carry out tasks for which they should not be authorized by virtue of the vulnerability, which may also affect the application's functionality.

System Resources: Should the application rely on JWTs to regulate access to system resources, an exploit of the vulnerability could jeopardize those resources.   
  
Reputation: If a security breach results from this vulnerability, the standing of the business or organization running the program may suffer.

**Evidence**

Provide a step by step guide on how to reproduce the vulnerability with screenshots

**Step 1.**

**Go to doubtfire-deploy/doubtfire-api/Gemfile.lock - Line 184**

**Step 2.**

**A screenshot of a computer program

Description automatically generated**

**Remediation Advice**

**The following steps can be taken to fix this issue.**

Update the JSON-JWT Feature: Make sure the json-jwt gem is up to date on your system. For any security updates or fixes that address this vulnerability, check the gem's repository.   
  
Use a Different Library: To avoid being impacted by this vulnerability, think about utilizing a different library or JWT verification technique. Seek for libraries that have a solid security record and are well-maintained.   
  
Validate JWTs Correctly: Make sure that your application is correctly validating JWTs if you are unable to upgrade the json-jwt gem right away. This involves employing a safe technique and key to accurately validate the cryptographic signature.   
  
Monitor for Suspicious Activity: Keep an eye out for any strange activity connected to JWT validation errors in your application's and server logs. This can assist you in identifying and thwarting possible attacks.

**References**

**json-jwt GitHub Repository:** Checking the latest versions or any security advisories <https://github.com/nov/json-jwt>

**OWASP JSON Web Token Cheat Sheet:** Provides guidance on securely implementing JWTs: <https://cheatsheetseries.owasp.org/cheatsheets/JSON_Web_Token_for_Java_Cheat_Sheet.html>

**online tutorials** - <https://youtu.be/c1sQVXU5PBM?si=Np6cM97lQQz_lAoR>

**Contact Details**

Payas Paul

612payaspaul@gmail.com

**Pentest Leader Feedback.**

The lead will provide feedback to enact on.