

**Finding Name: Arbitrary Code Injection**

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

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

**Finding Description**

A high severity Arbitrary Code Injection vulnerability has been identified in **pdfjs-dist@2.14.305**, which is introduced by **ng2-pdf-viewer@9.1.5**. This vulnerability allows attackers to inject and execute arbitrary code within the context of the application using the affected library. The issue has been fixed in version 4.2.67 of **pdfjs-dist**.

**Risk Rating**  
Impact: severe

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**

1. **System Compromise**:
   * **Impact**: Attackers can execute arbitrary code, potentially leading to full control over the affected system.
   * **Business Consequence**: This can result in data breaches, unauthorized access to sensitive information, and loss of system integrity.
2. **Data Breach**:
   * **Impact**: Sensitive data stored or processed by the application could be exposed.
   * **Business Consequence**: Legal and regulatory penalties, loss of customer trust, and significant financial losses can occur.
3. **Operational Disruption**:
   * **Impact**: Systems could be taken offline or disrupted by the exploit.
   * **Business Consequence**: Downtime can affect business operations, leading to lost revenue and productivity.
4. **Reputation Damage**:
   * **Impact**: Public disclosure of the vulnerability and exploitation could damage the company's reputation.
   * **Business Consequence**: Loss of customer confidence and potential impact on stock prices (for publicly traded companies).

**Affected Assets**

1. **Web Applications**:
   * Any web application using **ng2-pdf-viewer** with the vulnerable **pdfjs-dist** version is at risk.
2. **Servers and Infrastructure**:
   * Servers hosting the affected applications are vulnerable to compromise and resource hijacking.
3. **User Data**:
   * Data processed or stored by the affected applications could be exposed or altered.
4. **Business Processes**:
   * Critical business processes relying on the affected applications may be disrupted.

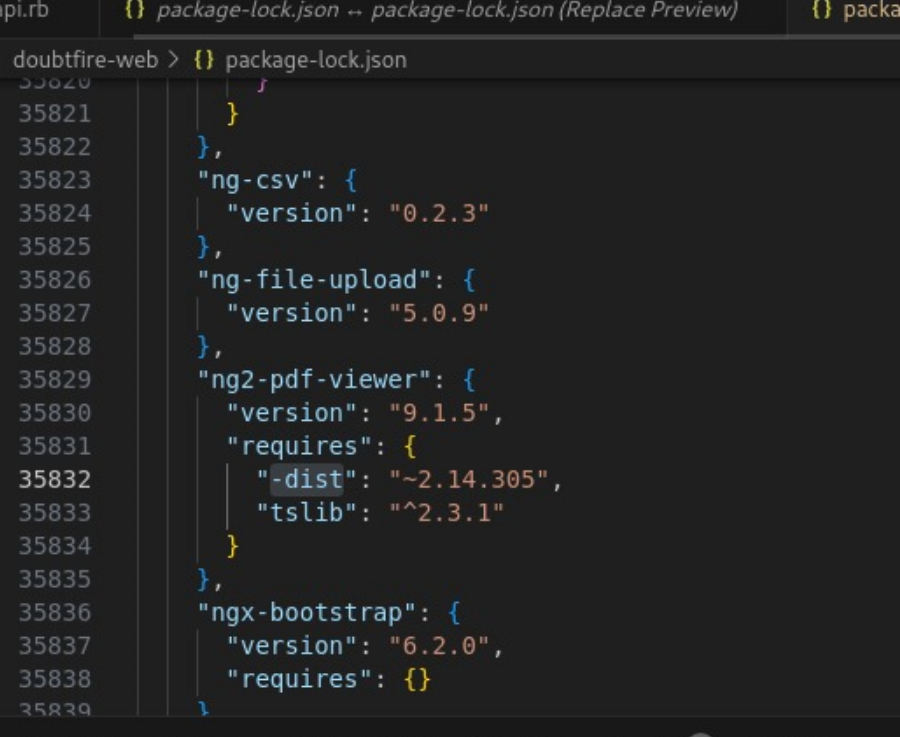
.

**Evidence**

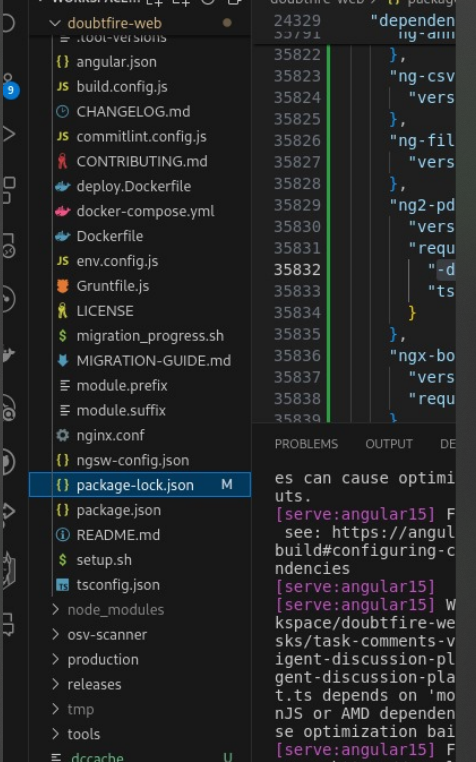
Provide a step by step guide on how to reproduce the vulnerability with screenshots

Step 1. Go to doubtfire-deploy/doubtfire-web/Package.json

Step 2.



Step 3.



Etc..

**Remediation Advice**

**Mitigation Strategies**

1. **Upgrade**:
   * Update **ng2-pdf-viewer** to a version that uses **pdfjs-dist@4.2.67** or later, where the vulnerability is fixed.
2. **Input Validation**:
   * Ensure that PDFs are validated and sanitized before processing.
3. **Monitoring**:
   * Implement enhanced monitoring to detect and respond to potential exploitation attempts.
4. **Patch Management**:
   * Regularly update and patch all dependencies to protect against known vulnerabilities.

**References**

Used Synk .

* OWASP Foundation. (2021). Code Injection. [online] Available at: https://owasp.org/www-community/attacks/Code\_Injection [Accessed 19 May 2024].
* Snyk. (2024). Arbitrary Code Injection (High Severity) in pdfjs-dist@2.14.305. [online] Available at: <https://security.snyk.io/vuln/SNYK-JS-PDFJSDIST-6810403> [Accessed 19 May 2024].
* Symantec Corporation. (2022). Internet Security Threat Report. [online] Available at: https://symantec.com/security-center/threat-report [Accessed 19 May 2024].

**Contact Details**

Payas Paul

2110994780

**Pentest Leader Feedback.**

The lead will provide feedback to enact on