

**Finding Name:** File path not properly sanitized

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Team** | **Role** | **Project** | **Quality Assurance** | **Is this a re-tested Finding?** |
| Charanpreet Singh | SRC | Project Co-Lead | Ontrack | Jaspriya and Payas |  |

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

**Finding Description**

The original code in the sanitized\_path method in file\_helper.rb file is intended to construct a safe file path by stripping leading and trailing whitespace from each path component and replacing all non-alphanumeric, non-hyphen, and non-parenthesis characters with underscores. However, this method does not adequately prevent directory traversal attacks, as it allows sequences like ../ which can lead to accessing directories and files outside of the intended scope.

**Risk Rating**  
Impact: Major  
Likelihood: Unlikely

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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. **Unauthorized Access**: The vulnerability in the original code could allow an attacker to access files and directories outside of the intended directory structure, potentially leading to unauthorized access to sensitive data.
2. **Data Breach**: If an attacker can read files outside of the application's directory, they may gain access to configuration files, source code, logs, or other sensitive information that could result in a data breach.
3. **System Compromise**: The ability to traverse directories may also enable an attacker to write or modify files, leading to system compromise or the introduction of malicious code.
4. **Regulatory Non-Compliance**: Accessing or exposing sensitive data may violate data protection regulations, leading to legal penalties and loss of customer trust.
5. **Service Disruption**: Exploiting this vulnerability could lead to denial of service if critical system files are accessed or altered.

**Affected Assets**

* **File System**: The file system where the application is hosted is directly affected, as the vulnerability can lead to unauthorized read or write operations outside of the application's directory.
* **Sensitive Data**: Any sensitive data stored on the server, such as personal user data, credentials, or proprietary information, is at risk if directory traversal is successful.
* **Application Integrity**: The integrity of the application is at stake if an attacker can modify the application's files or plant malicious code.

**Evidence**

**Step 1. The vulnerable piece of code can be found in FileHelper module in ‘file\_helper.rb’ file.**

A computer screen shot of text

Description automatically generated

**Remediation Advice**

Code that can prevent directory traversal

def sanitized\_path(\*paths)

safe\_paths = paths.map do |path\_name|

path\_name.strip.tap do |name|

# Remove directory traversal sequences

name.gsub!(/\.\.\/?/, '')

# Replace all non alphanumeric, hyphen, or period characters with underscore

name.gsub!(/[^\w\.\-]/, '\_')

end

end

# Join the paths and ensure the resulting path does not escape the intended directory

File.join(safe\_paths).gsub(/\/{2,}/, '/')

end

**Directory Traversal Removal:** name.gsub!(/\.\.\/?/, '') removes directory traversal sequences (.. or ../) from the path. This is a crucial security measure to prevent attackers from navigating to directories outside the intended scope.

**References**

* [**https://owasp.org/www-community/attacks/Path\_Traversal**](https://owasp.org/www-community/attacks/Path_Traversal)
* <https://wiki.owasp.org/index.php/Path_Traversal>

**Contact Details**

Name: Charanpreet Singh

Teams Name: CHARANPREET CHARANPREET SINGH

University Email: [s221538173@deakin.edu.au](mailto:s221538173@deakin.edu.au)

**Pentest Leader Feedback.**

The lead will provide feedback to enact on.