Absolute Path Traversal due to incorrect use of `send_file` call

Critical **onlaj** published **GHSA-g78x-q3x8-r6m4** on Apr 29

Package

onlaj/Piano-LED-Visualizer (pip)

Affected versions	Patched versions
<=1.3	>1.3

Description

A path traversal attack (also known as directory traversal) aims to access files and directories that are stored outside the web root folder. By manipulating variables that reference files with “dot-dot-slash (../)” sequences and its variations or by using absolute file paths, it may be possible to access arbitrary files and directories stored on file system including application source code or configuration and critical system files. This attack is also known as “dot-dot-slash”, “directory traversal”, “directory climbing” and “backtracking”.

Root Cause Analysis

The `os.path.join` call is unsafe for use with untrusted input. When the `os.path.join` call encounters an absolute path, it ignores all the parameters it has encountered till that point and starts working with the new absolute path. Please see the example below.

```
>>> import os.path
>>> static = "path/to/mySafeStaticDir"
>>> malicious = "/../../../../../etc/passwd"
>>> os.path.join(t,malicious)
'/../../../../../etc/passwd'
```

Since the "malicious" parameter represents an absolute path, the result of `os.path.join` ignores the static directory completely. Hence, untrusted input is passed via the `os.path.join` call to `flask.send_file` can lead to path traversal attacks.

In this case, the problems occurs due to the following code :

Piano-LED-Visualizer/webinterface/views_api.py
Line 970 in 6a732ca

```
970     return send_file("../Songs/" + value, mimetype='application/x-csv', attachment_filenam
```

Here, the `value` parameter is attacker controlled. This parameter passes through the unsafe `os.path.join` call making the effective directory and filename passed to the `send_file` call attacker controlled. This leads to a path traversal attack.

Proof of Concept

The bug can be verified using a proof of concept similar to the one shown below.

```
curl --path-as-is 'http://<domain>/api/change_setting?
second_value=no_reload&disable_sequence=true&value=../../../../../../../etc/passwd"'
```

Remediation

This can be fixed by preventing flow of untrusted data to the vulnerable `send_file` function. In case the application logic necessiates this behaviour, one can either use the `flask.safe_join` to join untrusted paths or replace `flask.send_file` calls with `flask.send_from_directory` calls.

A patch with a fix can be found in #351

References

- OWASP Path Traversal
- github/securitylab#669
- #351
- #350

This bug was found using *CodeQL by Github*

Severity

Critical 9.9 / 10

CVSS base metrics

Attack vector — Network

Attack complexity	**Low**
Privileges required	**None**
User interaction	**None**
Scope	**Changed**
Confidentiality	**High**
Integrity	**Low**
Availability	**Low**

CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:L/A:L

CVE ID

CVE-2022-24900

Weaknesses

CWE-73