Path Traversal due to `send_file` call in clinical-genomics/scout

0

Valid Reported on Mar 20th 2022

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. It should be noted that access to files is limited by system operational access control (such as in the case of locked or in-use files on the Microsoft Windows operating system).
This attack is also known as “dot-dot-slash”, “directory traversal”, “directory climbing” and “backtracking”.

Root Cause Analysis

Passing untrusted input to `flask.send_file` can lead to path traversal attacks.

Proof of Concept

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

```
curl --path-as-is -i -s -k 'http://<domain>/remote/static/unindexed?file=/.
```

This bug was found using *CodeQL by Github*

Occurrences

views.py L58-L67

```
curl --path-as-is -i -s -k 'http://<domain>/remote/static?
file=/../../../../../../../../..//etc/passwd'
```

views.py L70-L75

Here, the `file` parameter is attacker controlled and is used as the filename passed to the `send_file` call, this leads to a path traversal attack.

CVE
CVE-2022-1554
(Published)

Vulnerability Type
CWE-36: Absolute Path Traversal

Severity
Medium (6.8)

Visibility
Public

Status
Fixed

Found by

porcupineyhairs
@porcupineyhairs
unranked

Fixed by

Chiara Rasi
@northwestwitch
maintainer

This report was seen 1,185 times.

We are processing your report and will contact the clinical-genomics/scout
hours. 8 months ago

porcupineyhairs modified the report 8 months ago

We created a GitHub Issue asking the maintainers to create a `SECURITY.md` 8 months ago

Daniel Nilsson modified the report 8 months ago

Daniel Nilsson modified the report 8 months ago

Daniel Nilsson 8 months ago Maintainer

Thank you for reporting! The issue occurs on pages requiring flask authentication, and deployed behind IP filtering reverse proxy, so it is not in itself a remote information disclosing issue. But it does allow IP-listed, 2fa authenticated users to potentially read more than was intended. And could potentially become more sever together with other problems.

We were embarrassingly aware of this (see e.g. https://github.com/Clinical-Genomics/scout/issues/3128), but a clean fix requires a bit of refactoring. Thank you kindly for helping out; we will credit you as well for nudging us to fix it!

A clinical-genomics/scout maintainer has acknowledged this report 8 months ago

porcupineyhairs 8 months ago Researcher

@maintainer I just cloned a fresh copy of the repo and ran `docker compose up` . I can confirm that the bug can be exploited without authentication. I used this PoC

```
curl --path-as-is 'http://127.0.0.1:8000/remote/static/unindexed?
file=../../../../../../../etc/passwd'
```

porcupineyhairs 8 months ago Researcher

@maintainer Any ETA on the patch?

Daniel Nilsson 8 months ago Maintainer

Approximately a week. We have a refactor PR that addresses the issue, but it needs review and testing before merging.

Regarding your example, we quite agree regarding local dev instances. It does give /etc/passwd from the docker instance spun up. The docker vm files are normally not very
generically generated. The issue does not allow context switching on its own.
should also be behind secure authentication, if on a prod setup, and the whole web server
behind IP filtering

behind IP filtering.

porcupineyhairs 8 months ago — Researcher

@maintainer, CVE severity does not take third party firewall and other security measures into account. For example, let's say we have a RCE in a app. Now even though we may decide to place the node running the app off the network, since the app exposes itself on the network by default, a bug which exploits the app via a network interface would get a CVE rating corresponding to a network exploit and not a local exploit.

In this case, since by default, the bug in the app allows all external users access, the severity should be corresponding to that default scenario. An important thing to note here is, if it been the case that the app by default allows only a particular IP's to access, then the severity would be lower as you claim. However, given the defaults here at this time allow full access, I think it would be just to raise the severity back to the one I proposed.

porcupineyhairs 8 months ago — Researcher

@maintainer Any updates here?

porcupineyhairs 7 months ago — Researcher

@maintainer any updates here?

Chiara Rasi 7 months ago — Maintainer

Hello! The security issue should have been fixed now with this pull request: https://github.com/Clinical-Genomics/scout/pull/3303

Thanks again for the analysis!

Chiara Rasi validated this vulnerability 7 months ago

porcupineyhairs has been awarded the disclosure bounty ✔

The fix bounty is now up for grabs

The researcher's credibility has increased: +7

Chiara Rasi marked this as fixed in **4.52** with commit **952a2e** 7 months ago

Chiara Rasi has been awarded the fix bounty ✔

This vulnerability will not receive a CVE ✖

views.py#L58-L67 has been validated ✔

views.py#L70-L75 has been validated ✔

porcupineyhairs 7 months ago Researcher

@maintainer Thanks for the quick fix.

@admin can you please issue a cve now?

Jamie Slome 7 months ago Admin

Before we proceed with a CVE, I just want to establish if the maintainers are happy to assign and publish one.

@northwestwitch - are you happy for us to assign and publish a CVE for this report?

Chiara Rasi 7 months ago Maintainer

@admin yes thanks, we'd like to publish a CVE for this problem.

Jamie Slome 7 months ago Admin

Sorted 👍 It should be published within the next hour or two :)

CVE-2022-1554

Sign in to join this conversation