Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal') in django-s3file

Critical **codingjoe** published **GHSA-4w8f-hjm9-xwgf** on Jun 6

Package

django-s3file (pip)

Affected versions	Patched versions
< 5.5.1	5.5.1

Description

Impact

It was possible to traverse the entire AWS S3 bucket and in most cases to access or delete files.
The issue was discovered by the maintainer. There were no reports of the vulnerability
being known to or exploited by a third party, before the release of the patch.

If the `AWS_LOCATION` setting was set, traversal was limited to that location only.
If all your files handling views (like form views) require authentication or special permission, the thread is limited to privileged users.

Patches

The vulnerability has been fixed in version 5.5.1 and above.

Workarounds

There is no feasible workaround. We must urge all users to immediately updated to a patched version.

Detailed attack vector description

An attacker may use a request with malicious form data to traverse the entire AWS S3 bucket and perform destructive operations.

An attack could look as follows:

```
curl -X POST -F "s3file=file" -F "file=/priviliged/location/secrets.txt" https://www.example.com
```

This will result in a request with files set and opened:

```
>>> request.FILES.getlist("file")
[File("/priviliged/location/secrets.txt")]
```

Since this behavior is injected via a middleware, any view can be called this way and will carry any files defined by the attacker.

Via the `s3file` form field, any input name can be specified, including multiple inputs. For each input, multiple files can be freely
picked of the S3 bucket.

Scenarios and their practicality

There are four scenarios that would be considered practical in most setups:

1. Illegal file injection,
2. file deletion,
3. file retrieval & tree traversal.
4. code injection & remote code execution.

File deletion

An attacker knows the location of a privileged file, like a static asset. Next, the file is injected into a form view. The upload to function will move the file to a new location. This is effectively deleting the file, since the previous references to it are invalid, and will cause S3 to return a 404. Furthermore, the new location is unknown to the site operator.

File retrieval & tree traversal

An attacker knows the URL of a secret file and injects it into a form view. The view will move the file to a public location, making it accessible to the attacker. Since most form views will not be rate limited, this could also be used to guess files and traverse the file tree.

Illegal file injection

An attacker uses any form to upload a file to the temporary upload location. Next, the attacker injects that file into a request, does not validate the contents or is not equipped to handle the mime type. The latter could be used as a potential DOS vector.

In practice, this is not a practical risk in most hardened setup. Files should always be sanitized before processing, since files can be included in a request even without this security issues.

For more information

If you have any questions or comments about this advisory:

- Open an issue on GitHub
- Email us at johannes@maron.family

Severity

Critical

CVE ID

CVE-2022-24840

Weaknesses

CWE-22 CWE-96

Credits

 tunecrew

 syphar

 herrbenesch

 codingjoe