Improper Path Validation Allows Recursive Deletion of the Nginx Configuration Directory

Moderate **0xJacky** published **GHSA-m8p8-53vf-8357** 3 days ago

Package

/0xJacky/nginx-ui (Go)

Affected versions

< = 2.3.3

Patched versions

v2.3.4

Description

Summary

The nginx-ui configuration improperly handles URL-encoded traversal sequences. When specially crafted paths are supplied, the backend resolves them to the base Nginx configuration directory and executes the operation on the base directory (/etc/nginx). In particular, this allows an authenticated user to remove the entire `/etc/nginx` directory, resulting in a partial Denial of Service.

Details

The file deletion logic fails to correctly validate and normalize paths containing URL-encoded traversal sequences such as `..%252F`.

When such input is processed, the internal path resolution logic attempts to clamp the path into the allowed configuration directory. Instead of rejecting the traversal attempt, the clamping mechanism resolves the path to the base Nginx configuration directory itself.

Because the deletion handler invokes `os.RemoveAll`, which recursively removes directories, this results in the deletion of the entire `/etc/nginx` directory.

This behavior creates a dangerous interaction between path normalization and deletion logic:

- Traversal sequences are not rejected.

- Double-encoding (`..%252F`) is used to bypass initial shallow filters.
- The clamping mechanism resolves malicious paths to the base configuration directory.
- The deletion handler recursively deletes the resolved path.

As a result, an attacker can trigger deletion of the entire Nginx configuration directory instead of being blocked by path validation logic.

Root Cause

The vulnerability results from a combination of design flaws:

- **Improper Path Canonicalization**: URL-encoded traversal sequences are not properly rejected.
- **Unsafe Fallback Logic**: The `GetConfPath` clamping mechanism returns the base configuration directory when traversal is detected instead of rejecting the request.
- **Unsafe Deletion Primitive**: The deletion handler invokes `os.RemoveAll`, which recursively deletes directories without additional safeguards. (delete.go)

```
    // Delete the file or directory
    err = os.RemoveAll(fullPath)
    if err != nil {
        cosy.ErrHandler(c, err)
        return
    }
```

This interaction causes the deletion operation to target the most sensitive directory when a traversal attempt occurs.

Environment

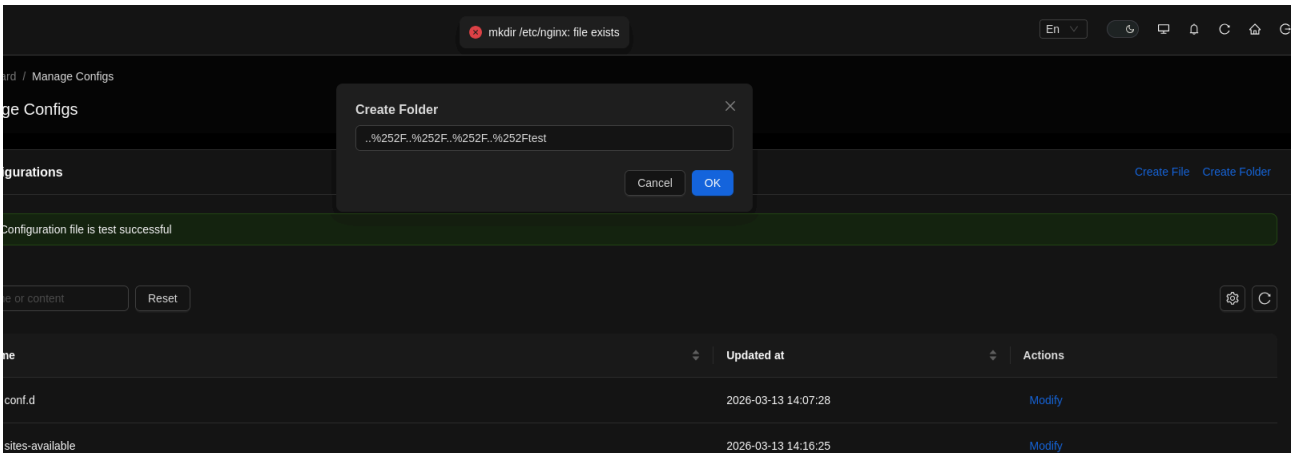
- **Server OS**: Kali Linux 6.17.10-1kali1 (6.17.10+kali-amd64)
- **Nginx UI Version**: nginx-ui v2.3.3
- **Deployment**: Docker / Default installation

Proof of Concept

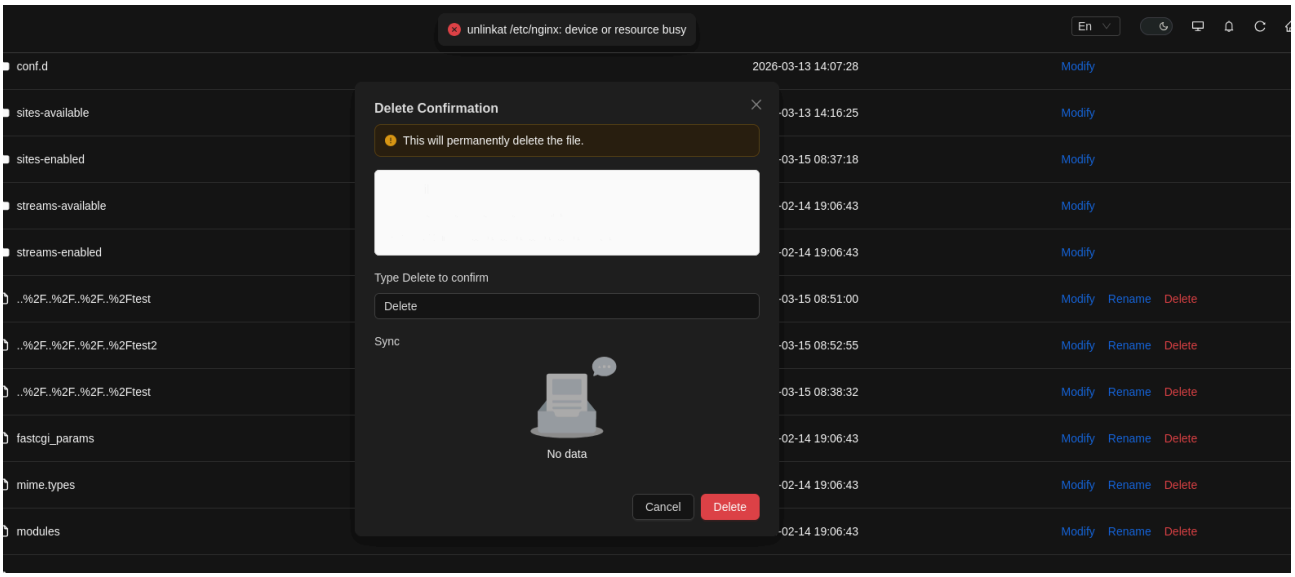
Steps to Reproduce

1. Log into nginx-ui.

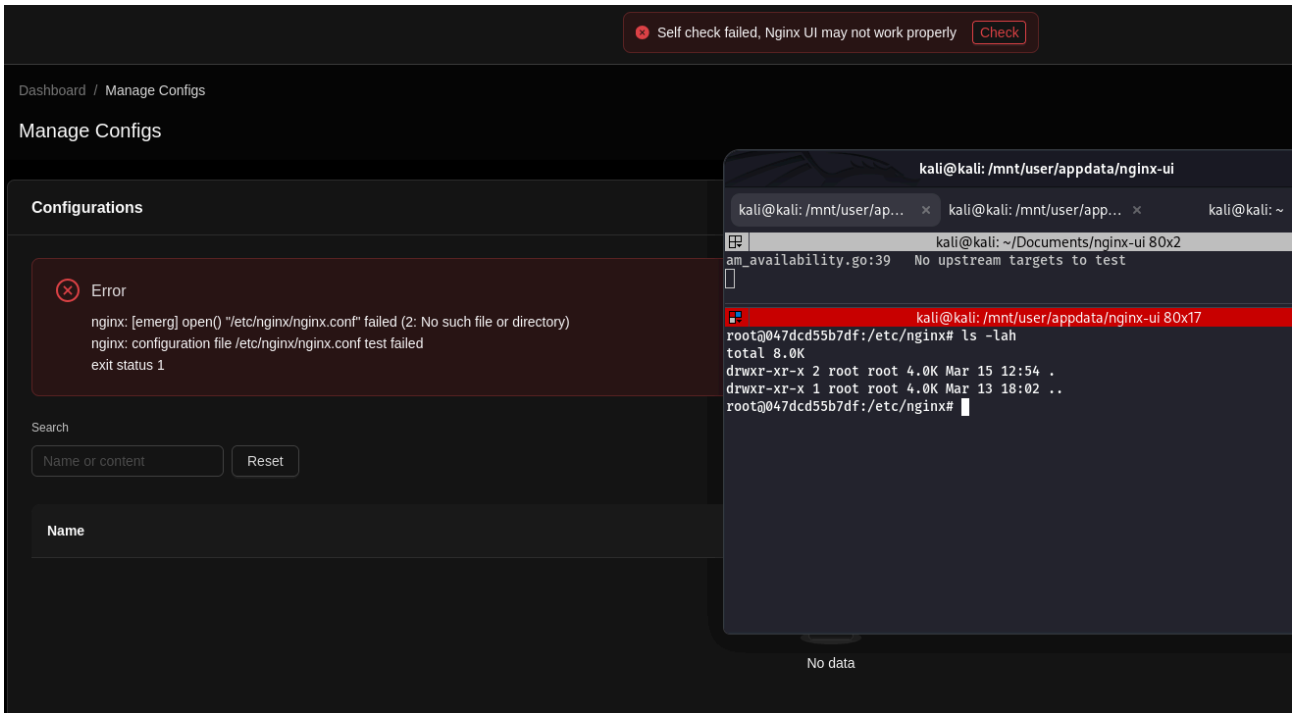
2. Go to Manage Configs and create a Folder named *..%252F..%252F..%252F..%252Ftest*



3. Observe that the backend resolves the path to /etc/nginx..
4. Now lets create a file called *testing*.
5. Save it and rename it to *..%252F..%252F..%252F..%252Ftest* (It is not possible to create it directly with the payload name so we have to rename it)
6. Go back to manage configs and Click Delete to remove the file we just created.
7. Check that there is an error:



8. Reload the website and check that the /etc/nginx folder has been completely removed:



Impact

An authenticated user capable of invoking the configuration deletion endpoint can trigger the recursive deletion of the entire Nginx configuration directory (`/etc/nginx`).

This results in:

- Immediate failure of the Nginx service due to missing configuration files.
- Loss of all Nginx configuration managed by nginx-ui.
- Denial of Service for all web services relying on the affected Nginx instance.

As the deletion operation uses a recursive filesystem call, the entire configuration directory is removed, leaving the system unable to restart Nginx until the configuration is manually restored.

Severity

Moderate 6.9 / 10

CVSS v4 base metrics

Exploitability Metrics

Attack Vector	Network
Attack Complexity	Low
Attack Requirements	None
Privileges Required	High
User interaction	None

Vulnerable System Impact Metrics

Confidentiality	None
Integrity	None
Availability	High

Subsequent System Impact Metrics

Confidentiality	None
Integrity	None
Availability	None

Learn more about base metrics

CVSS:4.0/AV:N/AC:L/AT:N/PR:H/UI:N/VC:N/VI:N/VA:H/SC:N/SI:N/SA:N

CVE ID

CVE-2026-33027

Weaknesses

- CWE-22
- CWE-73

Credits

dapickle Reporter