

Arbitrary File Write via Archive Extraction (Zip Slip)

Affecting [github.com/u-root/u-root/pkg/tarutil](#) package, versions <0.9.0

INTRODUCED: 1 SEP 2020 CVE-2020-7669 CWE-22 FIRST ADDED BY SNYK

Share

How to fix?

Upgrade [github.com/u-root/u-root/pkg/tarutil](#) to version 0.9.0 or higher.

Overview

[github.com/u-root/u-root/pkg/tarutil](#) is a package that provides Go versions of standard Linux tools and bootloaders. It also provides tools for compiling Go programs in a single binary and creating initramfs images.

Affected versions of this package are vulnerable to Arbitrary File Write via Archive Extraction (Zip Slip). It is vulnerable to both leading and non-leading relative path traversal attacks in tar file extraction.

PoC

```
package main import ( "os" "fmt" tar "github.com/u-root/u-root/pkg/tarutil" ) func main() { file := "relative.tar" f, err := os.Open(file) if err != nil { fmt.Println(err) } defer f.Close() if err := tar.ExtractDir(f, &quot;.&quot;); err != nil { fmt.Println(err) } }
```

with "relative.tar" being a tar archive that includes a file with filepath that uses leading or non-leading "../" .

Details

It is exploited using a specially crafted zip archive, that holds path traversal filenames. When exploited, a filename in a malicious archive is concatenated to the target extraction directory, which results in the final path ending up outside of the target folder. For instance, a zip may hold a file with a "../file.exe" location and thus break out of the target folder. If an executable or a configuration file is overwritten with a file containing malicious code, the problem can turn into an arbitrary code execution issue quite easily.

The following is an example of a zip archive with one benign file and one malicious file. Extracting the malicious file will result in traversing out of the target folder, ending up in /root/.ssh/ overwriting the authorized_keys file:

```
+2018-04-15 22:04:29 ..... 19 19 good.txt
+2018-04-15 22:04:42 ..... 20 20 ../../../../../../root/.ssh/authorized_keys
```

References

- [GitHub Commit](#)
- [GitHub PR](#)
- [GitHub PR](#)

PRODUCT

Snyk Open Source

Snyk Code

Snyk Container

Snyk Infrastructure as Code

Test with Github

Test with CLI

RESOURCES

Vulnerability DB

Documentation

HIGH

Search by package name or CVE

Snyk CVSS

Exploit Maturity Proof of concept

Attack Complexity Low

Integrity HIGH

See more

> NVD 7.5 HIGH

Do your applications use this vulnerable package?

In a few clicks we can analyze your entire application and see what components are vulnerable in your application, and suggest you quick fixes.

Test your applications

Snyk Learn

Learn about Arbitrary File Write via Archive Extraction (Zip Slip) vulnerabilities in an interactive lesson.

Start learning

Snyk SNYK-GOLANG-
ID GITHUBCOMUROOTUROOTPKGRTARUTIL-570428

Published 1 Sep 2020

Disclosed 1 Sep 2020

Credit Georgios Gkitsas of Snyk Security Team

Report a new vulnerability

Found a mistake?

[Disclosed Vulnerabilities](#)

[Blog](#)

[FAQs](#)

COMPANY

[About](#)

[Jobs](#)

[Contact](#)

[Policies](#)

[Do Not Sell My Personal Information](#)

CONTACT US

[Support](#)

[Report a new vuln](#)

[Press Kit](#)

[Events](#)

FIND US ONLINE

TRACK OUR DEVELOPMENT



© 2022 Snyk Limited

Registered in England and Wales. Company number: 09677925

Registered address: Highlands House, Basingstoke Road, Spencers Wood, Reading, Berkshire, RG7 1NT.