Path Traversal in gruntjs/grunt

1

Valid Reported on Jan 26th 2022

Description

Grunt is a JavaScript task runner, a tool used to automatically perform frequent tasks such as minification, compilation, unit testing, and linting. In GruntJS, file.copy operations in GruntJS are not protected against symlink traversal for both source and destination directories.

Scenario 1 - Restricted File Read

If a local attacker has write access to the source directory of file.copy, they can create a symlink to a restricted file. When the source directory is then copied from, either by the root user or a GruntJS task / cronjob running as root, the symlink is resolved and the contents of the restricted file will be copied to the destination directory with default umask permissions rw-r--r--, the directory will also be copied with permissions drwxr-xr-x. allowing them to read the restricted file.

Proof of Concept

1: As a lower-privileged user:

```
mkdir src
ln -s /etc/shadow src/shadow
```

2: As root execute the following PoC

```
grunt = require('grunt')
grunt.file.copy("src", "dest")
```

3: The lower privileged user can read the contents of the /etc/shadow file in the dest directory

```
cat dest/shadow
```

Scenario 2 - Restricted File Write

If an attacker has write access to both the source directory and the destination directory (if it has already been created) of file.copy, they can create a symlink to a restricted file in the destination directory and a file of the same name in the source directory. When the destination directory is then copied to, either by the root user or a cronjob running as root, the symlink is resolved to a restricted file and the file of the same name in source is copied to the resolved file path of the symlink in destination

Proof of Concept

1: As a lower-privileged user:

```
mkdir src
mkdir dest
ln -s /etc/shadow2 dest/shadow2
echo "<overwrite shadow file here>" > src/shadow2
```

2: As root execute the following PoC

```
grunt = require('grunt')
grunt.file.copy("src", "dest")
```

3: The /etc/shadow2 file is overwritten

```
<overwrite shadow file here>
```

Comparison with cp command

The standard cp command on all Linux systems copies the symlink object in directories instead of resolving it.

Impact

If a local attacker has write access to the source directory and read access to the directory containing the destination directory, they are able to abuse the file.copy ope
restricted files such as /etc/shadow which contains all the hashed passwords of users on the Linux system, they can they escalate their privileges by cracking the password or even SSH

Linux system, they can they escalate their privileges by cracking the password or even SSH private keys. If an attacker has write access to the source and destination directories, they are able to abuse the file.copy operation to overwrite restricted files such as /etc/shadow with their own shadow file and replace the root password with their own or even sign their own pair of SSH keys and replace the SSH public key with their own, guaranteeing them to escalate their privileges.

Recommended Fix

For directories, the file.copy should copy the symlink object rather than resolve it just like the standard cp command on Linux systems. Additionally, if a file in a destination directory is a symlink, then it should not be overwritten so as to prevent unintended consequences.

Occurrences

JS file.js L295L308

CVE
CVE-2022-0436
(Published)

Vulnerability Type
CWE-22: Path Traversal

Severity
High (7.1)

Visibility
Public

Status
Fixed

Found by

haxatron
@haxatron
pro

Fixed by

Vlad Filippov
@vladikoff
maintainer

This report was seen 1,015 times.

We are processing your report and will contact the **gruntjs/grunt** team within 24 hours. 10 months ago

haxatron modified the report 10 months ago

haxatron modified the report 10 months ago

haxatron modified the report 10 months ago

haxatron modified the report 10 months ago

We created a **GitHub Issue** asking the maintainers to create a `SECURITY.md` 10 months ago

haxatron modified the report 10 months ago

haxatron modified the report 10 months ago

haxatron modified the report 10 months ago

haxatron modified the report 10 months ago

haxatron modified the report 10 months ago

Vlad Filippov validated this vulnerability 10 months ago

haxatron has been awarded the disclosure bounty ✔

The fix bounty is now up for grabs

We have sent a fix follow up to the **gruntjs/grunt** team. We will try again in 7 days. 10 months ago

We have sent a second fix follow up to the **gruntjs/grunt** team. We will try again in 10 days. 10 months ago

We have sent a third and final fix follow up to the **gruntjs/grunt** team. This report is now

considered stale. 9 months ago

Jamie Slome 8 months ago Admin

@haxatron - I have reached out to the maintainer for this. Have you had any thoughts on a patch?

haxatron 8 months ago Researcher

I can try to, give me a week to do so.

Jamie Slome 8 months ago Admin

Sure 👍

haxatron 8 months ago Researcher

I just wanted to update here that this is currently quite tough to do as it is difficult to prevent race conditions in symlink removal / creation, but if we want to we can interface GruntJS's copy to ShellJS's copy. (We can also replace the rimraf dependency with ShellJS's rm as well). But I'll leave that up to the maintainer.

haxatron 8 months ago Researcher

^^^ What I mean by the above is to use a dependency (ShellJS) which is proven safe.

Vlad Filippov 8 months ago Maintainer

I have added a fix based on your suggestions here: The fix landed in https://github.com/gruntjs/grunt/pull/1740/files and was published in v1.5.0 of Grunt: https://www.npmjs.com/package/grunt

We can also explore adding what shelljs does in these cases...

Jamie Slome 8 months ago Admin

Thanks, Vlad! Are you able to `mark as fixed` using the dropdown section belo

haxatron 8 months ago Researcher

I can confirm that the fix resolves the issue when src is a symlink, but it does not resolve the issue when dest is a symlink. In that case, if dest is a symlink, do not follow the symlink, but you should instead overwrite it.

As for ShellJS, ShellJS cp is a direct implementation of the cp command on Linux, so it might be better to use that.

Vlad Filippov 8 months ago Maintainer

Thanks for the review, I shall mark this as fixed once I address the issue with "dest" symlinks...

Appreciate the input from everyone on this

haxatron 8 months ago Researcher

```
internal/fs/utils.js:269
    throw err;
    ^

Error: ENOENT: no such file or directory, symlink '../src/shadow' -> 'dest/shadow/shac
    at Object.symlinkSync (fs.js:1095:3)
    at Object.file._copySymbolicLink (/root/node_modules/grunt/lib/grunt/file.js:474:1
    at copy (/root/node_modules/grunt/lib/grunt/file.js:298:10)
    at /root/node_modules/grunt/lib/grunt/file.js:305:7
    at Array.forEach (<anonymous>)
    at Object.copy (/root/node_modules/grunt/lib/grunt/file.js:304:29)
    at Object.<anonymous> (/root/grunt/test.js:2:12)
    at Module._compile (internal/modules/cjs/loader.js:999:30)
    at Object.Module._extensions..js (internal/modules/cjs/loader.js:1027:10)
    at Module.load (internal/modules/cjs/loader.js:863:32) {
  errno: -2,
  syscall: 'symlink',
  code: 'ENOENT',
  path: '../src/shadow',
  dest: 'dest/shadow/shadow'
}
```

I get the above when running POC 1.

Is https://github.com/gruntjs/grunt/blob/main/lib/grunt/file.js#L469 correct? destpath is already dest/shadow, joining with the basename will make it dest/shadow/shadow.

Vlad Filippov 8 months ago — Maintainer

I will double check...

Vlad Filippov 8 months ago — Maintainer

Hey @haxatron, could you take a look at https://github.com/gruntjs/grunt/pull/1743 is that what you were suggesting to handle for dest paths?

haxatron 7 months ago — Researcher

Yes that is correct, though I see potential issues where an attacker can create another symlink right after it has been deleted but just before it is written to.

Vlad Filippov 7 months ago — Maintainer

I merged the requested fixes as part of https://github.com/gruntjs/grunt/releases/tag/v1.5.2 Release v1.5.2 is available on GitHub and NPM now.

Vlad Filippov marked this as fixed in **1.5.2** with commit **aad3d4** 7 months ago

Vlad Filippov has been awarded the fix bounty ✔

This vulnerability will not receive a CVE ✘

file.js#L295L308 has been validated ✔

Jamie Slome 7 months ago — Admin

Great work @haxatron & @vladikoff ♥

Sign in to join this conversation