Technical Advisory – Apple macOS XAR – Arbitrary File Write (CVE-2022-22582)

Rich Warren March 15, 2022

```
Vendor: Apple
Vendor URL: https://www.apple.com/
Systems Affected: macOS Monterey before 12.3, macOS Big Sur before 11.6.5 a
Author: Richard Warren <richard.warren[at]nccgroup[dot]trust>
Advisory URLs: https://support.apple.com/en-us/HT213183, https://support.ap
CVE Identifier: CVE-2022-22582
Risk: 5.0 Medium CVSS:3.1/AV:L/AC:L/PR:L/UI:R/S:U/C:N/I:H/A:N
```

Summary

In October 2021, Apple released a fix for CVE-2021-30833. This was an arbitrary file-write vulnerability in the `xar` utility and was due to improper handling of path separation (forward-slash) characters when processing files contained within directory symlinks.

Whilst analysing the patch for CVE-2021-30833, an additional vulnerability was identified which could allow for arbitrary file-write when unpacking a malicious XAR archive using the `xar` utility.

Impact

An attacker could construct a maliciously crafted `.xar` file, which when extracted by a user, would result in files being written to a location of the attacker's choosing. This could be abused to gain Remote Code Execution.

Details

Following the patch of CVE-2021-30833, files containing a forward-slash within the name property would be converted to a `:` character instead, as shown in the screenshot below:

As mentioned in the previous advisory, when attempting to extract a `.xar` file which contains both a directory symlink and a directory with the same name, an error is encountered, as the directory is created before the symlink.

However, after some experimentation, it was noted that `xar` processes the Table of Contents (TOC) backwards, this is demonstrated in the following example.

First, we create a `.xar` file with a TOC containing three entries – a, b, and c:

When listing the contents, we can see that the symlink directory ‘c’ is processed first:

This means that putting a directory symlink *before* the real directory (i.e., first from the top-down) within the TOC would cause it to fail with the message shown previously – since `xar` will refuse to create a symlink if a directory with the same name already exists – at which point it will skip over the symlink creation and just write the file to the real directory instead.

However, if we put the symlink directory at the end of the TOC, this will cause the symlink directory creation to *succeed* but the real-directory creation to *fail* – but, crucially, `xar` continues execution anyway, creating the file within our newly-created symlink-directory.

In summary, this means if we create a TOC with a symlink directory at the end, and a directory containing a file at the beginning, we can cause xar to:

1. First create the symlink directory
2. Try to create the directory (and fail, but continue)
3. Write the file into our symlink directory (achieving arbitrary file-write)

The following is an example of a TOC which exploits this vulnerability:

Now when extracting this file, we can see that the file `/tmp/test` is created successfully:

Recommendation

Upgrade to macOS Monterey 12.3, macOS Big Sur 11.6.5, macOS 10.15 Security Update 2022-003, or later.

Vendor Communication

```
2021–10–28 – Reported to Apple.
2022–03–14 – macOS 12.3, 11.6.5 and Security Update 2022–003 released, and
```

```
2022–03–15 – NCC Group advisory published.
```

About NCC Group

NCC Group is a global expert in cybersecurity and risk mitigation, working with businesses to protect their brand, value and reputation against the ever-evolving threat landscape. With our knowledge, experience and global footprint, we are best placed to help businesses identify, assess, mitigate & respond to the risks they face. We are passionate about making the Internet safer and revolutionizing the way in which organizations think about cybersecurity.

Published date: 2022-03-15

Written by: Richard Warren