CVE-2019-11510

Arbitrary file reading vulnerability in Pulse Connect Secure

Martin Pretz, 7060026

May 11, 2021

Abstract

1 Introduction

2 Description

The CVE-2019-11510 is a vulnerability that allows attackers to get arbitrary file reading access after they have send a special URI.[1]

This vulnerability has been assigned the maximum CVSS classifier of 10.0 Critical[2] **TODO: Warum dieser CVSS Score?**

This vulnerability is part of the CWE-22 class which is associated with "Path Traversal". That means that by explicitly specifying an abnormal path that is "[...] intended to identify a file or directory [...]" [3] it is possible to gain access to that file or directory without owning the required rights. This is made possible because the software uses an externally specified path and due to the way the software proceeds with that path. The effect is that the software resolves the given paths to files or folders that lie outside the resitriced directory.[3]

3 Threat Analysis

Regarding the STRIDE threat model, this vulnerability can be classified as an *information disclosure* in the first place, because primarily the attacker may read files unauthorized. But in the second place this vulnerability also leads to an *elevation of privileges* [4] because admin credentials are stored in plain-text inside an unrestricted file and because an attacker can arbitrary read files he can easyly obtain the admin credentials.[5] Even though one could argue that *information disclosure* "only" affects confidentiality and privacy, this vulnerability justifiably was classified with the CVSS 10.0 Critical, because the *elevation of privileges* additionally affects every single protective goal. **This includes the confidentiality, integrity, availability, authenticity, identify, deniability and privacy.** Thus every protective goal is lost an the attacker has complete control of the system and the stored data.[4]

3.1 Confidentiality

3.2 Integrity

3.3 Availability

3.4 Authenticity

3.5 Identify

3.6 Deniability

3.7 Privacy

4 Path traversal

Allthough path traversal was already mentioned in the previous chapter, one must take a closer look at path traversal in order to fully understand how it works. This will be done during the course of this chapter.

As previously mentioned path traversal attacks are used to access files or directories that lie outside the web root folder. This is usually achieved by using a path sequence containing the so called "dot-dot-slashes" (../) and its variations. Even though the attacker has access to the file system, he is also limited by the operating system's access control (meaning locked or in-use files).[6]

5 How can this vulnerability be exploited?

As already mentioned in 2, in order to exploit the vulnerability the attacker must send a request (e.g. via HTTP) to the target server that contains a path sequence used for path traversal (see chapter 4) and a special URI for the file that the attacker want to gain access to.[5]

"When a user logs into the admin interface of the VPN [...]"[5], the password is stored as plain-text within a MDB file (Microsoft Access Database). The corresponding file can be found at */data/runtime/mtmp/lmdb/dataa/data.mdb*. Since the attacker already has arbitrary access to all files of the system he can easyly obtain the admin password. With this information the attacker could perform further attacks, e.g. exploiting the CVE-2019-

11508, which allows an attacker to upload harmful files while using the credentials he obtained beforehand, since the credentials actually belong to an authanticated user.[5]

In other cases Kevin Beaumont reported that this vulnerability has lead to

6 Example usage

TODO: hier beispiel wie der exploit durchgeführt wird (mit programm)

7 Affected

All versions from between 8.2 to 8.2R12.1, 8.3 to 8.3R7.1 and 9.0 to 9.0R3.4 are affected.[1]

TODO: Länderliste

8 Prevention

Regarding the prevention of a such a path traversal attack, multiplie measures can be effectiv to different extents. Obviously the best way to prevent this type of attack is to operate the system/program without expecting the user to input a path. But since there are applications that inevitably require the user's input, this challange has to be adressed in a different way.[6]

In those cases it is advised to make sure, that the user can not specify all parts of the path, instead embeding the user input self defined paths. Another way could be to normalize the path before proceeding eliminating all irregularities.[6]

Furthermore use indexing for language related files instead of actually entering the language name. For example, the programm should not expect the user to write the word "Czechoslovakian", instead it should provide the option to select this language via an index value like "5 = Czechoslovakian".[6]

Last but not least, the usage of chroot jails or code access policies is advised in order to resitrict where files can obtained or saved to.[6]

9 Detection

10 Solution

Since this vulnerability is caused by the unintended behaviour of Pulse Connect Secure while resolving provided paths and therefore the internal implementation of this software, the vulnerability has been removed by a software patch provided by Pulse Secure.

TODO: Alternativen?

11 Conclusion

References

[1] *Nvd - cve-2019-11510*, https://nvd.nist.gov/vuln/detail/CVE-2019-11510, (Accessed on 05/10/2021).

[2] *Pulse vpn vulnerability analysis (cve-2019-11510) — awake security*, https://awakesecurity.com/blog/pulse-vpn-vulnerability-analysis-cve-2019-11510/, (Accessed on 05/11/2021).

[3] *Cwe - cwe-22: Improper limitation of a pathname to a restricted directory ('path traversal') (4.4)*, http://cwe.mitre.org/data/definitions/22.html, (Accessed on 05/10/2021).

[4] D. J. Schneider, *It-sicherheit - dr juergen schneider - dhbw mannheim - angewandte informatik - ss 2021 - 2 - angriffe und schwachstellen*, 2021.

[5] *Cve-2019-11510: Proof of concept available for arbitrary file disclosure in pulse connect secure - blog — tenable*, https://de.tenable.com/blog/cve-2019-11510-proof-of-concept-available-for-arbitrary-file-disclosure-in-pulse-connect-secure, (Accessed on 05/10/2021).

[6] *Path traversal — owasp*, https://owasp.org/www-community/attacks/Path_Traversal, (Accessed on 05/10/2021).