Why path traversal attacks are so devasating but relatively easy to avoid in theory

CVE-2019-11510

Martin Pretz, 7060026

May 17, 2021

Abstract

1 Introduction

2 Description

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

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 [...]" [2] 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. [2]

3 Threat classification

3.1 STRIDE

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* [3] 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. [4]

3.2 CVSS Rating

The CVE-2019-11510 has been assigned the CVSS rating of *10.0 CRITCAL* based upon the version 3 system and the rating of *7.5 HIGH* within the version 2 system. [1] Regarding the CVSS 3.1 system, this vulnerability has the following attack vectors: *AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H.*

The first attack vector (AV) is specified as network (N), meaning that this vulnerability is so called "remotely exploitable" which means that the attacker does not have to be physically present at the target. The attack complexity (AC) is defined as low (L) which means that no special conditions or circumstances need to be present for this vulnerability to be exploited. The attacker can repeatedly use this vulnerability. Regarding the third attack vector (PR) it is defined as none (N) since no privileges are required because this attack is using regular requests. Furthermore this vulnerability requires no (N) user interaction (UI) to be exploited. The Scope (S) of the vulnerability is specified as changed (C) because the attack does not only affect the attacked component but indeed affects the whole system. The impact metrics confidentialy (C), integrity (I) and availability (A) are all highly (H) affected by this vulnerability.[5]

Even though one could argue that the attack type *information disclosure* "only" affects confidentiality and privacy with ragrds to STRIDE, the attack type *elevation of privileges* does affect every single protective goal. [3] Thus this paper concludes that the given CVSS rating is justifiably at *10.0 CRITCAL* and would rate this vulnerability the same way.

4 What is 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 can send a simple HTTP GET request to the target endpoint 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. [4]

"When a user logs into the admin interface of the VPN [...]" [4], 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. [4]

In different cases the attacker was able to read `etc/passwd` by sending the path `/dana-na/../dana/html5acc/guacamole/../../../../../../etc/passwd?/dana/html5acc/guacamole/` to the appropriate endpoint of the Pulse Secure VPN. [7]

Kevin Beaumont reported that this vulnerability was used to gain admin access, installing VNC using PsExec, disabling endpoint security and finally installing the Sodinokibi ransomware.[8]

6 Protective Goals

For all protective goals applies that even if it cannot be compromised in the first place it is only a quastion of when the attacker will be able to compromise the corresponding protective goal. Eihter directly by exploiting the CVE-2019-11510 or indirectly by exploiting any other vulnerabilities he may find.

6.1 Confidentiality

The Confidentiality of the data is not longer guaranteed as the attacker can arbitrary read files. Furthermore if there would be some sort of first line limitation against the file reading it would not stand long because the attacker can read the passwords and therefore obtain the admin password which grants presumably full system control.

6.2 Integrity

The integrity of the data is only ensure in the first place since this vulnerability "only" allows attackers to read files. But in the second place the integrity of the data is lost due to the fact that the attacker achieved a privilege escalation and is capable of changing files or uploading harmful changes either directly or by using another vulnerability.

6.3 Availability

The availability of the network is also no longer guaranteed as the attackers has admin access he can temporarily or even permanently paralyze the VPN.

6.4 Authenticity

It can also no longer be assumed that the data is authentic because the attacker has change/write permissions because of the admin access.

6.5 Identity

Because the attacker as admin in the network, the identity goal is lost. Additionally he may be able to obtain the passwords of other users which will then compromise every their user identity.

6.6 Deniability

Since the intruder has admin access he can erase all hints on his presence e.g. by deleting log files.

6.7 Privacy

Even though in theory the privacy within the VPN is no longer guaranteed it has to be noted that the attacker can only intecerpt traffic within the network in the first palce. Of course in the the end the attacker can also use the access he gained to attack different devices within the network and if successfully can also compromise the privacy on user devices within the network.

7 Example usage

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

8 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] Even though a patch was provided by Pulse Secure in April 2019, initially little attention was paid to this vulnerability until in August 2019 a proof of concept exploit was released which encouraged attackers to exploit this weakness. [8] At this time Bad Packets detected mass scanning activity from attackers who were looking for vulnerable endpoints. Subsequently Bad Packets scanned 41,580 Pulse Secure VPN endpoint from which 14,528 were vulnerable even three months after Pulse Secure provided a patch. [9]

Country	Vulnerable endpoints
United States	5.010
Japan	1.511
United Kingdom	830
Germany	789
France	626
Netherlands	420
Israel	406
Switzerland	307
Canada	296
South Korea	281
All Other Countries	4.052

Table 1: Number of vulnerable endpoints in August 2019 [9]

9 Mitigation

During the course of this chapter it will be discussed how this vulnerability can be mitigated. Thereby, the analysis is divided in three different aspects which are the implementation, the architecture and design, as well as the operation.

9.1 Implementation

One of the most important methods of prevention is the input validation. It is recommended to define acceptable input based on the "accept known good" validation strategy. So all input either must directly comply with the specifications or must be converted in a way that does comply. It is of crucial importance that the validation is performed as accurate and comprehensive as possible, otherwise the validation misses its point. More precisely, looking "[...] exclusively for malicious or malformed [paths]" [2] is not what should be done during validation. Instead try to limit directly what characters are allowed so that the attacker cannot even enter a harmful path. Concrete it should be prohibited to enter ".." character, instead only allow a single "." character. File extensions should also be concerned, meaning that the file extensions is not allowed to be included in the path but must be selected from a predefined list. This way it can be restricted what types of files can be read. Furthermore directory separators such as the the "/" or the "\" character should be prohibited. This has to be done not only once per path but multiple times in order to minimize the error rate. For example if the validation checks for the string "../" within the path ".../...//", there would still remain the "../" string, even tough "../" has been removed twice from the original path. [2]
Furthermore warnings or error messages should only contain as little information as possible that are exclusively useful to intended recipients. This way, attackers cannot obtain detailed information about the inner workings of the system. [2]

9.2 Architecture and Design

This section will summ up the most common decisions regarding the architecture and design of a programm in order to prevent or at least mitigate the impact of the vulnerability.

If any user input is to be validated on the client side, it is advised to ensure that this validation process is duplicated on the server side to prevent that the attacker can use CWE-602 vulnerabilities he makes use of CVE-2019-11510. This would allow the attacker to bypass client sided validations and thus the attacker would still be able to send an spcial path to perform a path traversal. [2]
When running the software in production, ensure that the program the lowest privilege level that is necessary to achieve the task. This way, even if an attacker sends an harmful to the server, the software which is proceessing the input is limited due to its low privileges. Though this method cannot prevent this attack from happening, it can limit the impact.[2]
Access to files can be prevented if they are stored outside the web applications root. However if this is not possible, a different solution would be to store files in a separate directory and to use the web servers access control features to restrict the access, preventing attackers from directly requesting files. For example, defining a fixed constant in each calling programm, checking if this constant does exist in the file and if absent the file was directly requested and should be closed. [2]
If user input is required and the number of possible inputs is limited, assign/map an ID to every possible input value accepting only the predefined IDs and rejecting requests using undefined IDs. [2] "For example, ID 1 could map to "inbox.txt" and ID 2 could map to "profile.txt"." [2] This method is called an "Acces Reference Map".

9.3 Operation

Content of this section include methods of prevention for operating the program in production environment.

The use of a dedicated application firewall can be effectiv to detect and prevent attacks trageting this vulnerability. This is escpecially useful, if the vulnerability cannot be fixed independently because the program belongs to a third party. [2] It has to be noted that this can only act as a temporary solution until a patch is available.

Additionally, the program can be run inside a sandbox environment in way that separates it from the

operating system. This will heavily restrict the accessable files and directories. But it has to be noted that this solution only reduces the impact an attack can have on the operating system, while the application itself can "[...] still be subject to compromise." [2] Furthermore the degree of effectiveness depends on the scope and possibilities of the sandbox. Eligible sandbox systems could be Unix chroot jail, SELinux or AppArmor. [2]

10 Detection

A fundamental method for detection is to review the source code with special attention to potential weaknesses that may facilitate path traversal attacks. This method is highly effective and can also be used for a focused review of certain code parts. Such an analysis can either be conducted manually, in which case it is recommended that at least two people are involved in the analysis to reduce the error rate as far as possible, or it can be performed automatically.

As allready mentioned in section 9.2, the access to files can be resitriced by defining constants for each calling program. But this method is not just about prevention, furthermore it can be observed if an file was unauthorized requested directly, providing an detection method for path traversal attacks.

11 Solution

This vulnerability has been fixed in with an out-of-cycle path in April 2019 by Pulse Secure. But even though the patch was provided the vulnerability was active until 2020 due to the fact that some users of Pulse Connect Secure VPN had stil not installed the patch.

Even though the specific patch can not be reviewed, it can be assumed that Pulse Secure has used one of the mitigation methods mentioned in 9 in order to fix the vulnerability.

12 Conclusion

Regarding the fix of the vulnerability, the hypothesis of the paper is that Pulse Secure has fixed the vulnerability by adjusting their implementation. Specifically it is assumed that they implemented some sort of validation that is capable of handling potentially harmful requests.

References

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

[2] *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).

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

[4] *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).

[5] *Nvd - cvss v3 calculator*, https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator, (Accessed on 05/17/2021).

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

[7] *Pulse secure 8.1r15.1/8.2/8.3/9.0 ssl vpn - arbitrary file disclosure (metasploit) - multiple webapps exploit*, https://www.exploit-db.com/exploits/47297, (Accessed on 05/17/2021).

[8] *Cve-2019-11510: Critical pulse connect secure vulnerability used in sodinokibi ransomware attacks - blog — tenable*, https://de.tenable.com/blog/cve-2019-11510-critical-pulse-connect-secure-vulnerability-used-in-sodinokibi-ransomware, (Accessed on 05/10/2021).

[9] *Over 14,500 pulse secure vpn endpoints vulnerable to cve-2019-11510 – bad packets*, https://badpackets.net/over-14500-pulse-secure-vpn-endpoints-vulnerable-to-cve-2019-11510/, (Accessed on 05/10/2021).