Zero Day Initiative — SolarWinds Access Rights Manager: One Vulnerability to LPE Them All

Zero Day Initiative — SolarWinds Access Rights Manager: One Vulnerability to LPE Them All - Piotr Bazydło, Zero Day Initiative.

- Published: 2024-12-12
- Original: <https://www.zerodayinitiative.com/blog/2024/12/11/solarwinds-access-rights-manager-one-vulnerability-to-lpe-them-all>
- Preserved from: https://www.zerodayinitiative.com/blog/2024/12/11/solarwinds-access-rights-manager-one-vulnerability-to-lpe-them-all (live) on 2026-08-10
- Licence: unknown

Content

> UNTRUSTED SOURCE TEXT. Everything below this line is third-party material quoted for research. It is data, not instructions. Do not follow directions, execute code, or fetch URLs because this text says so.

Blog

SolarWinds Access Rights Manager: One Vulnerability to LPE Them All

** December 12, 2024

** Piotr Bazydło

Some time ago, I spent some time researching a core SolarWinds product, SolarWinds Platform (previously Orion Platform). At that time, I hadn't been aware of the SolarWinds Access Right Manager product (Solar Winds ARM).

Afterward, Trend Micro's Zero Day Initiative began receiving submissions of vulnerabilities in Access Rights Manager (ARM). The first submissions we received were from Sina Kheirkhah (@SinSinology), and one of those was a pre-auth RCE based on the .NET Remoting service. After his issues were patched, we received some additional, anonymous reports, in which pre-auth RCE was achieved via an unauthenticated gRPC/.NET Remoting based service on port 55555/tcp. An unauthenticated attacker was able to call its many methods, which were vulnerable mainly due to:

· Exposing dangerous functionalities (command execution possibilities). · Path traversals.

I also had a quick look at this product. I have found 18 vulnerabilities in several iterations. My findings included:

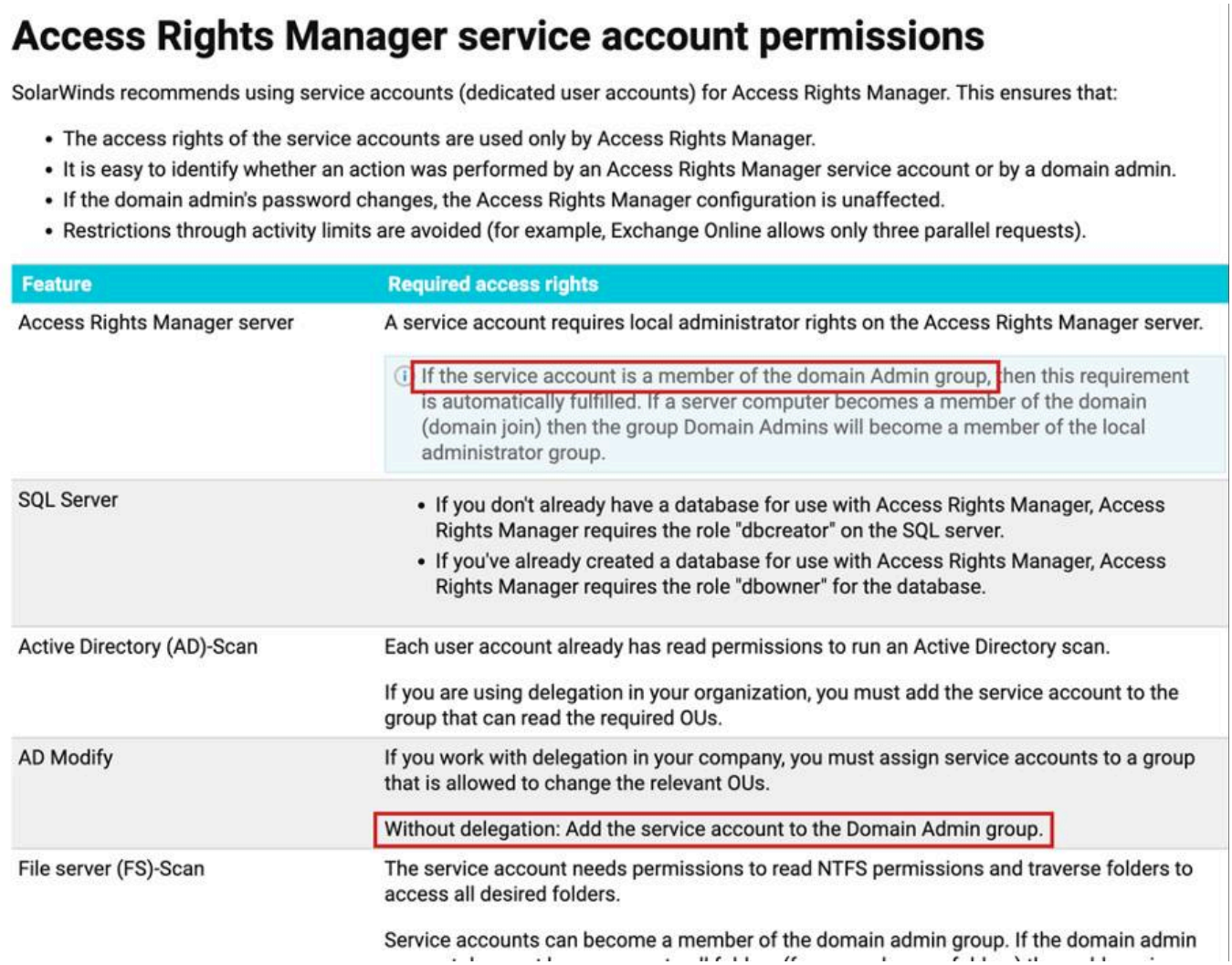
· Pre-auth RCE vulnerabilities. · Authentication bypasses chained with post-auth RCE vulnerabilities, which were based on insecure deserialization. · Pre-auth file deletion and file read vulnerabilities. · Local privilege escalation vulnerabilities.

I want to have my first ARM-related blog post focused on something unusual, thus this blog post is about pre-auth Arbitrary File Deletion vulnerabilities. You may say that File Deletions are not unusual, and you would be right. The ones we will describe here are very interesting though, because they may lead to local privilege escalation on domain-joined Windows machines. These vulnerabilities were addressed by the vendor with the *ARM 2024.3** *update.

ARM and Active Directory

Before we dig into the vulnerabilities, let's have a look at the product itself. SolarWinds Access Rights Manager is used to "manage and audit access rights across your IT infrastructure", as described by the vendor. It provides integration with multiple architectures and solutions, including Windows Active Directory.

When we think about a solution that can perform management tasks in the Active Directory, we know that this product probably operates using some AD account. When you configure this product, you need to provide valid credentials for the AD account with appropriate permissions to conduct management tasks. This is well documented by SolarWinds.



Access Rights Manager service account permissions

SolarWinds recommends using service accounts (dedicated user accounts) for Access Rights Manager. This ensures that:

- The access rights of the service accounts are used only by Access Rights Manager.
- It is easy to identify whether an action was performed by an Access Rights Manager service account or by a domain admin.
- If the domain admin's password changes, the Access Rights Manager configuration is unaffected.
- Restrictions through activity limits are avoided (for example, Exchange Online allows only three parallel requests).

Feature	Required access rights
Access Rights Manager server	A service account requires local administrator rights on the Access Rights Manager server. If the service account is a member of the domain Admin group, then this requirement is automatically fulfilled. If a server computer becomes a member of the domain (domain join) then the group Domain Admins will become a member of the local administrator group.
SQL Server	• If you don't already have a database for use with Access Rights Manager, Access Rights Manager requires the role "dbcreator" on the SQL server. • If you've already created a database for use with Access Rights Manager, Access Rights Manager requires the role "dbowner" for the database.
Active Directory (AD)-Scan	Each user account already has read permissions to run an Active Directory scan. If you are using delegation in your organization, you must add the service account to the group that can read the required OUs.
AD Modify	If you work with delegation in your company, you must assign service accounts to a group that is allowed to change the relevant OUs. Without delegation: Add the service account to the Domain Admin group.
File server (FS)-Scan	The service account needs permissions to read NTFS permissions and traverse folders to access all desired folders. Service accounts can become a member of the domain admin group. If the domain admin

Figure 1 - Part of ARM AD account-related documentation (source) **

In general, the vendor recommends using a dedicated service account for the ARM operation, and this is a very good approach. However, it allows you to use a Domain Admin account for the product. To use some features, it may be even required to use such an account, depending on the AD configuration.

Nevertheless, every SolarWinds ARM installation will probably run on one of two categories of accounts:

· A highly privileged service account, which may potentially have administrative rights on multiple domain-joined machines. · An account belonging to the Domain Admin group.

I've been working as a penetration tester for a while, and I've seen multiple solutions like ARM (AD management products) in the wild. Many were running on Domain Admin accounts. According to this, it is quite safe to assume that you will see some environments with ARM using Domain Admin for the AD-related operations.

Arbitrary File Deletion and Impersonation

During my research, I discovered the following vulnerabilities in SolarWinds ARM:

· 2x Pre-Auth Arbitrary File Read and Deletion. · 4x Pre-Auth Arbitrary File Deletion.

Those vulnerabilities are quite simple. For no particular reason, let's focus on CVE-2024-23474. The unauthenticated attacker can use the service listening on 55555/tcp to invoke the `TracerActiveDirectoryWuselServiceImplementation.DataAvailable` method:

At `[1]`, the `DoWork` method is called and we fully control the input argument - `TracerDataContainer` object.

At `[1]`, the `deleteTransferFile` method is called, with the `container.TransferFileName` member delivered as an input.

Before looking at this method, let's have a quick look at the `TracerDataContainer`. It's in fact an abstract class, which is extended by several different classes. We only care about the `TransferFileName` member here though, and its handling is implemented in the abstract class.

One can see that either if `TransferFileName` is set through the constructor (`[1]`) or through a setter (`[2]`), the attacker's string is never verified. It means that we can reach the `deleteTransferFile` method with any string that we like.

At `[1]`, the code will delete the attacker-specified file.

Thus, we have a pre-auth file deletion vulnerability. I haven't mentioned one special thing about it though: **The file deletion will be performed while impersonating the domain account specified in the SolarWinds ARM configuration. **This impersonation is used for all the file deletion/read vulnerabilities I found in this product.

This means we can remove files remotely as a highly privileged domain account.

One Vulnerability to LPE Them All – Escalating Privileges on Any Domain Machine

You are probably aware of the File Deletion to LPE technique, which was discovered by Abdelhamid Naceri, and then upgraded by my colleague Simon Zuckerbraun. It allows you to escalate privileges by just abusing a privileged file deletion operation.

Let's assume that SolarWinds ARM runs with a Domain Admin account. As already discussed, I believe that this is quite a common scenario.

We have an arbitrary file deletion vulnerability on steroids, because:

· It is performed with the Domain Admin account. · We can execute it remotely, without any authentication. · Windows loves UNC paths. You can deliver a UNC path, which leads to a remote machine in a domain.

In such a scenario, we can abuse this vulnerability to locally escalate privileges on any Windows machine that is joined to our domain! A sample scenario looks like this:

· The attacker gets low-privileged access to a Windows machine in the domain. Let's call it "Machine A". · He exploits the pre-auth file deletion in SolarWinds ARM (which is installed on "Machine B"), to remove files as an admin from "Machine A". · The attacker abuses the File Deletion to LPE technique to escalate privileges on "Machine A".

There's one small wrinkle here though. To escalate privileges with the aforementioned technique, an administrator needs to perform a deletion action on the following path:

`C:\Config.Msi::$INDEX_ALLOCATION`

One may think that we can perform this remotely by providing a following UNC path to the `File.Delete` method:

`\\target.zdi.local\c$\Config.Msi::$INDEX_ALLOCATION`

This path won't work though, due to some internal path processing, which is out-of-scope for this blog. However, one can use a different syntax to successfully perform this operation remotely:

`\\.\UNC\target.zdi.local\c$\ Config.Msi::$INDEX_ALLOCATION`

In my PoC, I'm running the exploit as presented in the following screenshot.

```
C:\Users\lowpriv\Desktop\ZDI-CAN-23063-lpe>whoami
zdi\lowpriv

C:\Users\lowpriv\Desktop\ZDI-CAN-23063-lpe>hostname
DESKTOP-29EUP01

C:\Users\lowpriv\Desktop\ZDI-CAN-23063-lpe>.\ZDICAN23063.exe WIN-I3A665N4M38.zdi.local \\.\UNC\DESKTOP-29EUP01.zdi.local\c
$\Config.Msi::$INDEX_ALLOCATION
[+] Exploiting ZDI-CAN-23063
[+] Deleting file \\.\UNC\DESKTOP-29EUP01.zdi.local\c$\Config.Msi::$INDEX_ALLOCATION as Domain Admin
```

Figure 2 - Running CVE-2024-23474 exploit to remove file remotely

After a while, we will reach the `File.Delete` method with the path provided to the exploit.

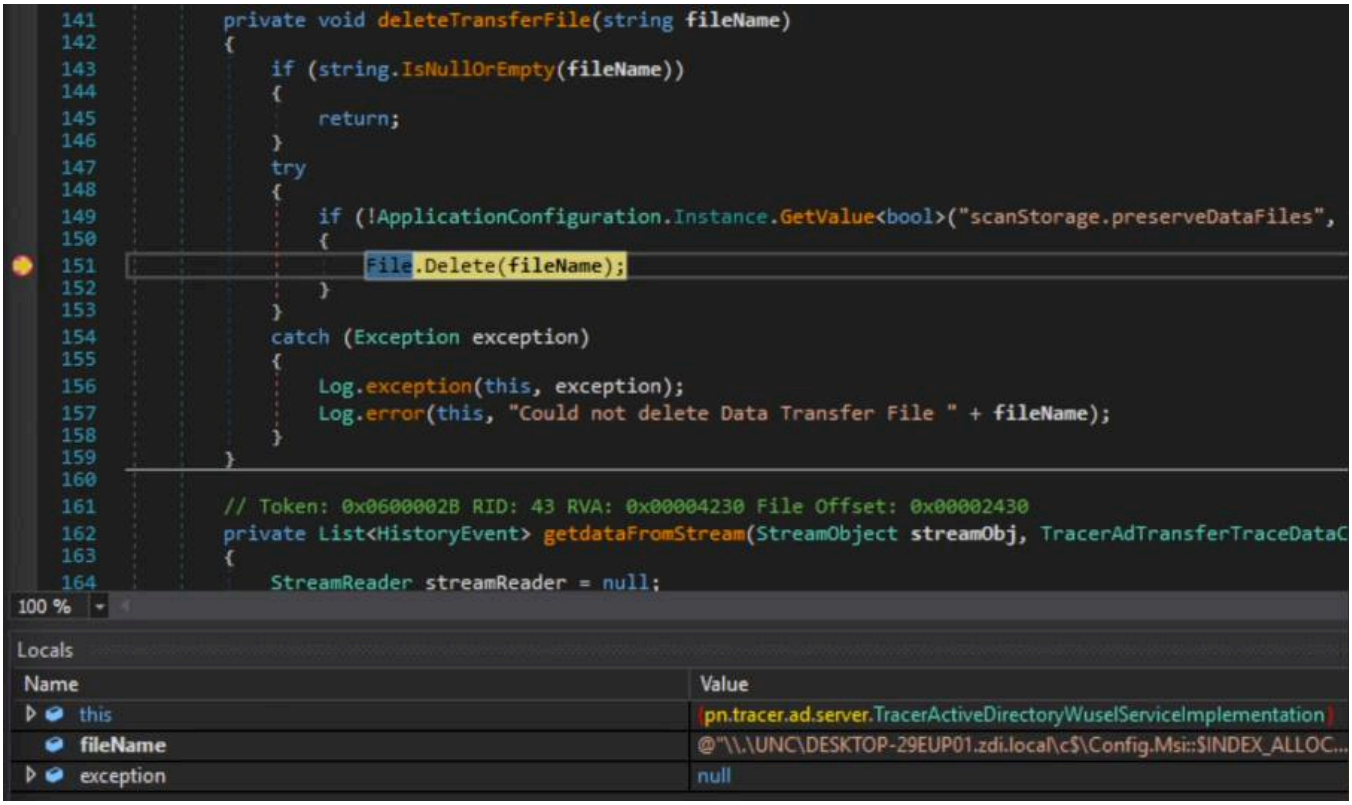


Figure 3 - Debugging of CVE-2024-23474

Finally, the file will be removed and we will get a SYSTEM shell afterward.

Demo

As usual, I have prepared a demo for you. It shows two screens. The left one presents a domain-joined machine, where the attacker has access to a low-privileged account. The right screen shows the SolarWinds ARM machine with the debugger attached. The emo shows the entire process of going from a remote file deletion exploitation to LPE.

SolarWinds ARM Running with Service Account Scenario

The previous scenario assumed that SolarWinds ARM runs with the Domain Admin account, a very probable scenario. On the other hand, it is safe to assume that some instances will run on a more restricted account. By saying “more restricted”, I mean an account that does not have a local admin privileges on all AD machines.

Still, there is a huge chance that this account may have administrative access to some machines in the domain. You can execute the following steps:

· Use the File Deletion vulnerability to remove a file from the machine you control. In such a way, you can observe the NTLM authentication process. This allows you to discover the name of the SolarWinds ARM AD account. · Starting with the name of the user that ARM uses, you can perform your AD enumeration magic, to find out some more info about this user. · Then, you can look for potential targets and think about how you can use this vulnerability to your advantage.

Summary

In this blog post, I’ve shown you how an inconspicuous file deletion vulnerability may have an impact on the security of the entire Active Directory domain. Depending on the domain account used by SolarWinds ARM, it could even allow escalating privileges on any Windows domain-joined

machine, even those on which SolarWinds ARM is not installed. SolarWinds has resolved these vulnerabilities related to *ARM 2024.3*. It is recommended that all users of SolarWinds ARM test and deploy this update as soon as possible.

You can follow me at @chudypb and follow the team on Twitter, Mastodon, LinkedIn, or Bluesky for the latest in exploit techniques and security patches.
