Local File Inclusion				Severity: High
CVSS v3.1	7.7	**Vector**	AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:N/A:N	
Privileges	SAS Developer			
Version	SAS 9.4 for predictive performance management extension			

Local file inclusion (LFI) is a vulnerability which allows an attacker to include files that are locally stored on the server. This kind of vulnerability allows the attacker to obtain access to information which may lead to more severe type of attacks up to a full server compromise.

This is achieved by exploiting a dynamic file inclusion mechanism which is not correctly implemented on the application. This allows the attacker to manipulate the input in order to access files which should not be visible.

The vulnerable resource:

- https://<application-baseurl>/SASStudio/sasexec/sessions/9f121b79-9148-4b47-bcc6-8afc845847ba/workspace/~~ds~~etc/passwd

Follow the steps below to replicate the results

The target application allows for file download. An attacker can manipulate the file path to download in order to include other files present on the target filesystem.

Below, an example of including the */etc/passwd* file.

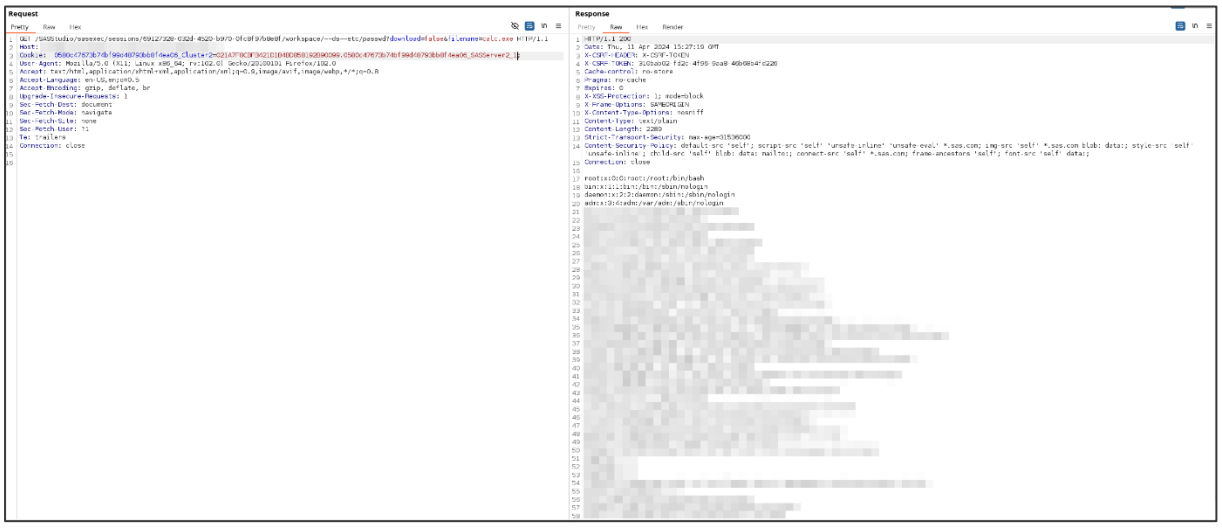

Figure 1: */etc/passwd* file inclusion.

This vulnerability also allows for directory listing which grants the attacker valuable information to enumerate files which can then be downloaded.

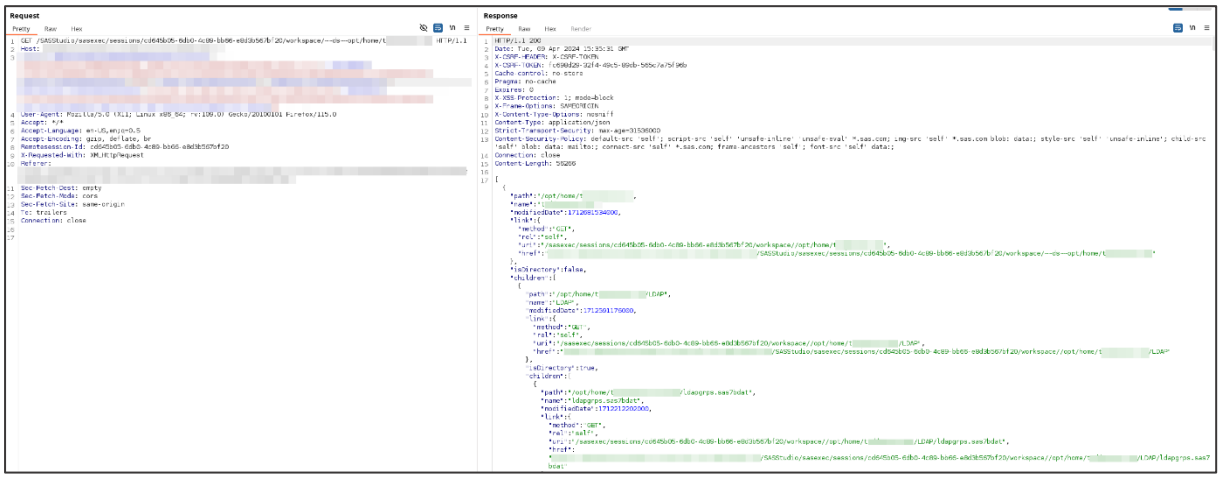

Figure 2: Directory listing for */opt/home/<user>*.

By combining the directory listing with the local file inclusion, it is possible to easily download the entire directories with their content.

In this example, a python script has been developed in order to download the content of the */opt/home/<user>* directory.

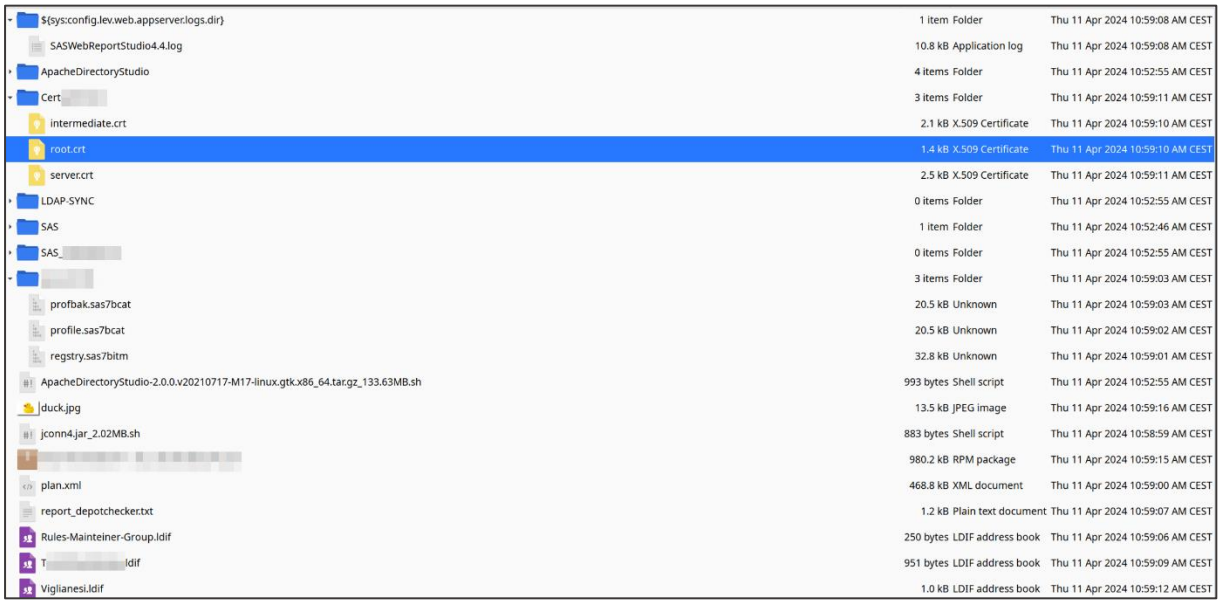

Figure 3: Downloaded content for the directory */opt/home/<user>*.

REMEDIATION:

In order to prevent Local File Inclusion, it is advisable to:

- If you need dynamic path concatenation, ensure you only accept required characters such as "a-Z0-9" and do not allow ".." or "/" or "%00" (null byte) or any other similar unexpected characters.
- Allow inclusion only from a directory and directories below it. This ensures that any potential attack cannot perform a directory traversal attack.
- Ensure the user cannot supply all parts of the path – surround it with your path code.

For further information, please refer to:

- https://owasp.org/www-community/attacks/Path_Traversal