Link: https://cloudops-sisense.s3.us-east-2.amazonaws.com/fix_log4j_Sisense.zip

Sisense Windows Patch for Log4j Vulnerability - Updated December 23, 2021

The following document will walk you through applying an updated patch on your existing WIndows-based Sisense installations in order to address the log4j security vulnerability.

Introduction:

Question	Answer
What versions of Sisense on Windows does this patch support?	Patch is fully compatible with all 8.x and above versions of Sisense on Windows. Patch should work for all 7.x versions of Sisense as well. Please note that we are still assessing 6.x versions.
What approach was taken to tackling the log4j vulnerability?	The approach applied is to patch existing installations of Sisense windows-based deployments by removing the JNDI classes from the existing log4j libraries deployed as part of Sisense. This will prevent Log4j components from performing vulnerable JNDI Lookups

	that allow remote code execution. By removing these classes, the security vulnerability will be avoided. Note that Sisense does not use the JNDI classes and therefore will have no impact on the operating of the platform. Currently there is no plan to upgrade the log4j version that is used in Sisense in order to apply the security fix, and if so it will be taken into consideration only on new upcoming releases.
Which Sisense components are exposed and use log4j?	JVM Connectors - The java based connectors that are part of the Sisense deployment. CLR connectors are not affected. Shipper - The capability that is incorporated into Sisense which allows for the automatic shipping of log.
What does the script do?	The script will: 1. Stop Sisense windows services 2. Scan the connectors that are installed in Sisense for JNDI classes and remove them 3. Scan the shipper libraries and removes JNDI classes. 4. Start Sisense windows services

About the Script

Question	Answer
What does the patch package include?	fixLog4JSisense.ps1 - includes the commands to apply the changes in the patch. Logger.ps1 - includes the code to log the script.
Prerequisites to running patch	Administrator type user to run the script with access to the windows based server(s)
Testing Considerations	Patch was tested by Sisense and is certified to run on Sisense 8.x and above releases. After deployment, test the connectivity to data sources that are used in your builds and check that the shipper works.
Does the script generate a log?	The script will generate a log with a default location of “C:\Scripts\fix_log4j\log.txt" which will contain the output of what the script executes. Make sure no errors appear in the log to ensure the script runs properly.

	Another log that might be generated is going to be a powershell log that is defaulted to location “c:\Logs\PowerShellLog.log”. Note this log does not need to be reviewed.
Is there a prerequisite to running the script out of the box (meaning with no changes)?	7-Zip will need to be already installed on the Windows server, or the script will need to install it successfully. In this case location to the Internet to URL “https://7-zip.org/” is required.
Is there a way to rollback the script?	No, there is no way to rollback the changes. We suggest you backup the server before running the script in case Sisense will stop working after the patch is completed. However, note that the jar files that are modified are going to be backed up as part of the script or you can back them up manually as specified below in the script description. In case you did not backup you will need to reinstall Sisense.
Do I run this script this one time and I am covered, or might I need to run it again in the future?	The script should be run once. However, if new connectors are installed you will need to apply the same changes that script made. Of course if an upgrade to any of the existing Sisense releases you will need to run the script again.
What are the script assumptions?	The script assumes that the Sisense application was installed in: “C:\Program Files\Sisense\”

	Where the connector JARs that will be patched are located under ".\DataConnectors" (under Program Files and ProgramData) and the Shipper jar is located under "..\infra\Data\Shipper", which are the JARs that will be modified.

Running the Script

Step #	Instruction	Example
1	Unzip the patch package into the proper directory in which you wish to run the script from. Note the script is configured to run from the directory "C:\SCripts\fix_log4j". In order to make a chance to the location you will need to make changes to the script as follows: - Open "fixLog4JSisense.ps1" - Change the *$loggerPath* to the given location of the directory you unzipped the package to. - Change the location and name of the log to the appropriate folder by changing the *$LogPath.* - Save changes	Unzip "fix_log4j_Sisense" c:\

	- Note that	
2	Launch PowerShell (or any other capability that can run the script) with admin rights.	Run as administrator “Windows PowerShell”
3	If you are installing from in PowerShell, run the script “fixLog4jSisense.ps1”. Note you need to give the full path. Or launch the script from any appropriate capability.	C:\scripts\fix_log4j\fixLog4jSisense.ps1
4	Once the script completes, check the log to ensure that no errors were captured.	Open the log that was configured to be outputted.
5	Launch the Sisense site: - Test connectors used in the builds are working - If you use shipper, check that logs are shipped properly	n/a

Script (fixLog4jSisense.ps1) Explanation and Log Entries Walkthrough

Script Code Performing Action	Description	Log Output	Exceptions
# Prep	Sets the log location	“Starting”	n/a

Set-ExecutionPolicy Unrestricted -Force $loggerPath = "C:\Scripts\fix_log4j\Logger.ps1" . $loggerPath; $LogPath = "C:\Scripts\fix_log4j\log.txt" Write-Log -Level Info -Path $LogPath -Message "Starting"	and the path of the logging script.		
if (Test-Path "C:\Program Files\7-Zip\7z.exe")	Check to see if 7-zip installed and if not install 7-zip you would need to modify the script accordingly by changing to the proper utility.	"7-zip already Installed" OR "7-zip installed"	"Error Installing 7-zip Script stopped" - you would need to either have 7-zip installed. If another utility needs to be used the script will need to be changed accordingly.
[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12 $dlurl = 'https://7-zip.org/' + (Invoke-WebRequest	Download and install 7-zip from the 7-zip site.	"Installing 7zip"	n/a. If a failure occurs, install 7-zip manually or check to see if and why it is not installed properly.

-UseBasicParsing -Uri 'https://7-zip.org/' | Select-Object -ExpandProperty Links | Where-Object { ($_.outerHTML -match 'Download') -and ($_.href -like "a/*") -and ($_.href -like "*-x64.exe") } | Select-Object -First 1 | Select-Object -ExpandProperty href) $installerPath = Join-Path $env:TEMP (Split-Path $dlurl -Leaf) Invoke-WebRequest $dlurl -OutFile $installerPath Start-Process -FilePath $installerPath -Args "/S" -Verb RunAs -Wait Remove-Item $installerPath			
JVM Connectors			
Copy-Item -Path "C:\Program Files\Sisense\DataConnectors\JVMContainer" -Destination	Backup the existing connector jar. Note we assume the	“creating a backup for connectors”	“n/a” Doublecheck that that jar is blacked up properly to the

"C:\Program Files\Sisense\DataConnectors\JVMContainerBK" –Recurse	location of the application is as specified in this document before. You would need to change the script if this is not the case.		specified location.
n/a	Start handling the connector libraries.	“Start cleaning connectors”	n/a
Stop-Service -Name Sisense.JVMConnectorsContainer	The “JVMConnectorContainer” service is stopped. This is the service name that runs on 8.x and above versions of Sisense on Windows.	n/a	n/a
Get-ChildItem -Path "C:\Program Files\Sisense\DataConnectors\JVMContainer" -Filter "*.jar" -Recurse $output = &"C:\Program	Search through all the jar files and remove “jndilookup.class”.	“successfully scanned <name of the jar file>” For example:	“ERROR fixing from <name of the jar file>” - find out why the jar file was not modified. Check that the jar can be opened for modification.

Files\7-Zip\7z.exe" d $_.FullName org\apache\logging\log4j\core\lookup\JndiLookup.class 2>&1		INFO: successfully scanned C:\Program Files\Sisense\DataConnectors\JVMContainer\bin\connectorService.jar	
Start-Service -Name Sisense.JVMConnectorsContainer	Start the “JVMConnectorContainer” service.	n/a	n/a Double check that the service has started properly.
n/a	Indicate all connector jars were patched properly.	“Completed. please review the logs and make sure there are no ERRORs, if not connectors are clean”	"Error while handling connectors" - Check why the steps above might have failed. For example, the script has the wrong location of Sisense and/or the specified JAR, or service stop failed due to lack of permission to stop it.
Connectors from ProgramData			

Copy-Item -Path "C:\ProgramData\Sisense\DataConnectors" -Destination "C:\ProgramData\Sisense\DataConnectorsBK" –Recurse	Backup the existing ProgramData connector jar.	“creating a backup for connectors”	“n/a” Doublecheck that that jar is blacked up properly to the specified location.
n/a	Start handling the connector libraries.	“Start cleaning connectors”	n/a
Stop-Service -Name Sisense.JVMConnectorsContainer	The “JVMConnectorContainer” service is stopped. This is the service name that runs on 8.x and above versions of Sisense on Windows.	n/a	n/a
Get-ChildItem -Path "C:\ProgramData\Sisense\DataConnectors" -Filter "*.jar" -Recurse $output = &"C:\Program Files\7-Zip\7z.exe" d $_.FullName org\apache\logging\log4j\core\lo	Search through all the jar files and remove “jndilookup.class”.	“successfully scanned <name of the jar file>” For example: INFO: successfully scanned	“ERROR fixing from <name of the jar file>” - find out why the jar file was not modified. Check that the jar can be opened for modification.

okup\JndiLookup.class 2>&1		C:\ProgramData\Sisense\DataConnectors\MsSql\Sisense.MsSql.JVM.1.0.16423.10002.0.0\com.sisense.connectors.MsSql.jar	
Start-Service -Name Sisense.JVMConnectorsContainer	Start the "JVMConnectorContainer" service.	n/a	n/a Double check that the service has started properly.
n/a	Indicate all connector jars were patched properly.	"Completed. please review the logs and make sure there are no ERRORs, if not connectors are clean"	"Error while handling connectors" - Check why the steps above might have failed. For example, the script has the wrong location of Sisense and/or the specified JAR, or service stop failed due to lack of permission to stop it.
Shipper			

Copy-Item -Path "C:\Program Files\Sisense\infra\Data\Shipper" -Destination "C:\Program Files\Sisense\infra\Data\Shipper BK" –Recurse	Backup the shipper jars located in the specified path.	“creating backup for shipper”	n/a
n/a	Start handling the shipper libraries.	“start cleaning shipper”	n/a
Stop-Service -Name Sisense.Shipper	The “Shipper” service is stopped.	n/a	n/a
Get-ChildItem -Path "C:\Program Files\Sisense\infra\Data\Shipper" -Filter "*.jar" -Recurse $output = &"C:\Program Files\7-Zip\7z.exe" d $_.FullName org\apache\logging\log4j\core\lookup\JndiLookup.class 2>&1	Search through all the jar files and remove “jndilookup.class”.	“successfully scanned <name of the jar file>” For example: successfully scanned C:\Program Files\Sisense\infra\Data\Shipper\logstash-5.2.2\logstash-core\lib\com\fasterxml\jackson\core\jac	“ERROR fixing from <name of the jar file>” - find out why the jar file was not modified. Check that the jar can be opened for modification.

		kson-annotations\2.7.0\jackson-annotations-2.7.0.jar	
Start-Service -Name Sisense.Shipper	The “Shipper” service is started	n/a	n/a
n/a	Indicate all shipper jars were patched properly.	Completed. please review the logs and make sure there are no ERRORs, if not shipper are clean	“Error while handling shipper” - Check why the steps above might have failed. For example, the script has the wrong location of Sisense and/or the specified JAR, or service stop failed due to lack of permission to stop it.
Validation - Double Check cleanup was succesful			

Get-ChildItem -Path "C:\Program Files\Sisense\DataConnectors\JVMContainer" -Filter "*.jar" -Recurse -ErrorAction SilentlyContinue -Force | Foreach-Object { $found = $_.FullName; &"C:\Program Files\7-Zip\7z.exe" l $_.FullName org\apache\logging\log4j\core\lookup\JndiLookup.class | Select-String -Pattern "JndiLookup.class" -List | Foreach { echo $found $ErrorCount + 1 } } Get-ChildItem -Path "C:\ProgramData\Sisense\DataConnectors" -Filter "*.jar" -Recurse -ErrorAction SilentlyContinue -Force | Foreach-Object { $found = $_.FullName; &"C:\Program Files\7-Zip\7z.exe" l $_.FullName org\apache\logging\log4j\core\lookup\JndiLookup.class | Select-String -Pattern "JndiLookup.class" -List |	Validate that jars have been cleaned properly. If an entry is found, add it to the counter to record the number of issues found.	Update Ended Successfully	Update Had ERRORs

Foreach { echo $found $ErrorCount = 1 } } Get-ChildItem -Path "C:\Program Files\Sisense\infra\Data\Shipper" -Filter "*.jar" -Recurse -ErrorAction SilentlyContinue -Force | Foreach-Object { $found = $_.FullName; &"C:\Program Files\7-Zip\7z.exe" l $_.FullName org\apache\logging\log4j\core\lookup\JndiLookup.class | Select-String -Pattern "JndiLookup.class" -List | Foreach { echo $found $ErrorCount = 1 } }			