

A white rectangular object with blue lines

Description automatically generated

A white paper with blue lines

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

­­­­A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a chat

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

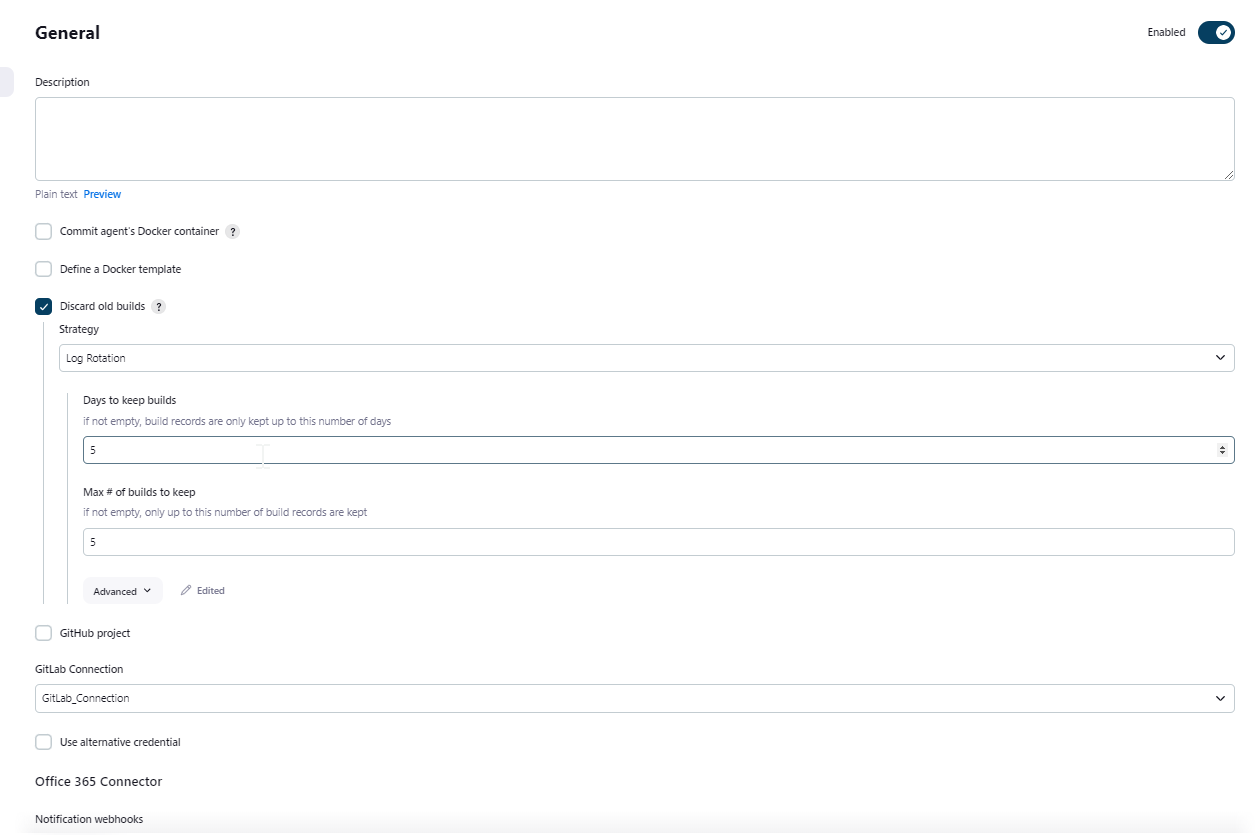
Description automatically generated

A screenshot of a computer

Description automatically generated

A close-up of a blank coupon

Description automatically generated



A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Set-ExecutionPolicy Bypass -Scope Process

$zipFilePath = "D:\security\_scan\dependency-check.zip"

$extractPath = "D:\security\_scan"

# Check if the destination folder already contains the extracted files

if (Test-Path (Join-Path $extractPath "dependency-check")) {

Write-Host "Files already extracted. Skipping the process."

} else {

# Unzip the file

Add-Type -AssemblyName System.IO.Compression.FileSystem

[System.IO.Compression.ZipFile]::ExtractToDirectory($zipFilePath, $extractPath)

Write-Host "Files extracted successfully."

}

A white rectangular object with a black border

Description automatically generated

set "reportsFolder=D:\security\_scan\reports"

if exist %reportsFolder% (

cd /d %reportsFolder%

if exist report.html (

del report.html

echo report.html deleted successfully.

) else (

echo report.html not found. Skipping deletion.

)

if exist report\*.zip (

del report\*.zip

echo report\*.zip deleted successfully.

) else (

echo report\*.zip not found. Skipping deletion.

)

) else (

echo Reports folder not found. Skipping deletion.

)

A close-up of a mail

Description automatically generated

cd /d D:\jenkins

set "serviceName=%Scan\_Components%"

if /i "%serviceName%" equ "WSO2MI" (

powershell -File "D:\security\_scan\ServiceExecutablePathMI.ps1" -serviceName %serviceName%

) else if /i "%serviceName%" equ "WSO2CARBON\_MI" (

powershell -File "D:\security\_scan\ServiceExecutablePathMI.ps1" -serviceName %serviceName%

) else (

powershell -File "D:\security\_scan\ServiceExecutablePath.ps1" -serviceName %serviceName%

)

A close-up of a computer screen

Description automatically generated

set "source=D:\security\_scan\reports\report.html"

set "destination=D:\security\_scan\reports"

set "zipFileName=report"

set "sevenZipPath=C:\Program Files\7-Zip"

REM Get the current date and time

for /f "delims=" %%a in ('wmic OS Get localdatetime ^| find "."') do set datetime=%%a

set "timestamp=%datetime:~0,4%-%datetime:~4,2%-%datetime:~6,2%\_%datetime:~8,2%-%datetime:~10,2%-%datetime:~12,2%"

REM Create the zip file with formatted timestamp

pushd "%destination%"

"%sevenZipPath%\7z.exe" a -tzip "%zipFileName%\_%timestamp%.zip" "%source%"

popd

echo "HTML file has been zipped successfully with formatted timestamp."

A white rectangular object with a black border

Description automatically generated

cd %WORKSPACE%

if exist "report\_\*" (

del /F /Q \*.\*

echo Folder deleted.

) else (

echo Folder does not exist. Skipping deletion.

)

xcopy /Y D:\security\_scan\reports\\*.zip .

echo Files copied

A screenshot of a computer

Description automatically generated

set folderPath=D:\security\_scan\report\_backup

if not exist "%folderPath%" (

mkdir "%folderPath%"

echo Folder created successfully.

) else (

echo Folder already exists.

)

copy D:\security\_scan\reports\\*.zip "%folderPath%"

powershell -File "D:\security\_scan\extract\_content.ps1"

A white rectangular object with black lines

Description automatically generated

@echo off

setlocal enabledelayedexpansion

set FILE\_PATH=D:\security\_scan\output.txt

set FILE\_CONTENT=

for /f "delims=" %%i in ('type %FILE\_PATH%') do (

set "FILE\_CONTENT=!FILE\_CONTENT!%%i --"

)

echo %FILE\_CONTENT% > fileContent.txt

endlocal

A screenshot of a computer

Description automatically generated

set "folderPath=D:\security\_scan"

if not exist "%folderPath%" (

mkdir "%folderPath%"

echo Folder created successfully.

) else (

echo Folder already exists.

)

A screenshot of a computer

Description automatically generated

Set-ExecutionPolicy Bypass -Scope Process

$zipFilePath = "D:\security\_scan\dependency-check.zip"

$extractPath = "D:\security\_scan"

# Check if the destination folder already contains the extracted files

if (Test-Path (Join-Path $extractPath "dependency-check")) {

Write-Host "Files already extracted. Skipping the process."

} else {

# Unzip the file

Add-Type -AssemblyName System.IO.Compression.FileSystem

[System.IO.Compression.ZipFile]::ExtractToDirectory($zipFilePath, $extractPath)

Write-Host "Files extracted successfully."

}

set "reportsFolder=D:\security\_scan\reports"

if exist %reportsFolder% (

cd /d %reportsFolder%

if exist report.html (

del report.html

echo report.html deleted successfully.

) else (

echo report.html not found. Skipping deletion.

)

if exist report\*.zip (

del report\*.zip

echo report\*.zip deleted successfully.

) else (

echo report\*.zip not found. Skipping deletion.

)

) else (

echo Reports folder not found. Skipping deletion.

)

cd /d D:\jenkins

set "serviceName=%Scan\_Components%"

if /i "%serviceName%" equ "WSO2MI" (

powershell -File "D:\security\_scan\ServiceExecutablePathMI.ps1" -serviceName %serviceName%

) else if /i "%serviceName%" equ "WSO2CARBON\_MI" (

powershell -File "D:\security\_scan\ServiceExecutablePathMI.ps1" -serviceName %serviceName%

) else if /i "%serviceName%" equ "tomcat8" (

powershell -File "D:\security\_scan\ServiceExecutablePathTomcat.ps1" -serviceName %serviceName%

) else if /i "%serviceName%" equ "tomcat7" (

powershell -File "D:\security\_scan\ServiceExecutablePathTomcat.ps1" -serviceName %serviceName%

)else if /i "%serviceName%" equ "dev7b\_file\_adapter" (

powershell -File "D:\security\_scan\ServiceExecutablePath\_File\_adapter.ps1" -serviceName %serviceName%

)else if /i "%serviceName%" equ "Tomcat7-109-medispan" (

powershell -File "D:\security\_scan\ServiceExecutablePathMedispan.ps1" -serviceName %serviceName%

)else (

powershell -File "D:\security\_scan\ServiceExecutablePath.ps1" -serviceName %serviceName%

)

set "source=D:\security\_scan\reports\report.html"

set "destination=D:\security\_scan\reports"

set "zipFileName=report"

set "sevenZipPath=C:\Program Files\7-Zip"

REM Get the current date and time

for /f "delims=" %%a in ('wmic OS Get localdatetime ^| find "."') do set datetime=%%a

set "timestamp=%datetime:~0,4%-%datetime:~4,2%-%datetime:~6,2%\_%datetime:~8,2%-%datetime:~10,2%-%datetime:~12,2%"

REM Create the zip file with formatted timestamp

pushd "%destination%"

"%sevenZipPath%\7z.exe" a -tzip "%zipFileName%\_%timestamp%.zip" "%source%"

popd

echo "HTML file has been zipped successfully with formatted timestamp."

cd %WORKSPACE%

if exist "report\_\*" (

del /F /Q \*.\*

echo Folder deleted.

) else (

echo Folder does not exist. Skipping deletion.

)

xcopy /Y D:\security\_scan\reports\\*.zip .

echo Files copied.

set folderPath=D:\security\_scan\report\_backup

if not exist "%folderPath%" (

mkdir "%folderPath%"

echo Folder created successfully.

) else (

echo Folder already exists.

)

copy D:\security\_scan\reports\\*.zip "%folderPath%"

powershell -File "D:\security\_scan\extract\_content.ps1"

# Set the file path

$filePath = "D:\security\_scan\output.txt"

# Read the content from the file

$fileContent = Get-Content -Path $filePath -Raw

# Replace "\n" with a new line character

$fileContent = $fileContent -replace "\\n", "`r`n"

# Set the output file path in the Jenkins workspace

$outputFilePath = Join-Path $env:WORKSPACE "fileContent.txt"

# Write the modified content to the output file

$fileContent | Set-Content -Path $outputFilePath

# Display the modified content

Write-Host "Modified content saved to $outputFilePath"