

**Backup & Restoration**

**Manual Using   
OneDrive**

**AWS CyberShift Initiative**

Date Prepared: 29th of June, 2023

**Contents Page**

[**Backup & Restoration Procedure 3**](#_gjdgxs)

[Windows 2022 AD Server 3](#_y2na05svqesb)

[Backup Steps 3](#_l9i8ynnupvho)

[Backing up the AD Server: 3](#_qyf74pqh5iji)

[Restore Steps 6](#_cy46q6nikl7k)

[Encryption Option 6](#_65tk27hhpyd6)

[Linux Web Server 7](#_y2na05svqesb)

[Prerequisites: Installing OneDrive 7](#_6nwb6k2yab87)

[Configure Sophos Automatic Backups 12](#_4hj0f9n9kril)

[Backup Steps 13](#_jjkhnatagaew)

[Creating a scheduled automatic backup: 14](#_mz4bsxln84u2)

[Restore Steps 15](#_qc4lw5yhzol4)

[**Security and Monitoring Software Backup 16**](#_o078llkbht7y)

[Windows AD Server Procedure 16](#_y2na05svqesb)

[Backup Locations 16](#_89ytpacicfib)

[Backup Steps 16](#_o5ma9i7wvm1i)

[Restore Steps 22](#_8e5tzsvkoh8g)

|  |
| --- |
| Backup & Restoration Procedure |
| Windows 2022 AD Server |
| Backing up the AD Server:  1. **Install the Windows Server Backup feature**  * Open Server Manager on your Windows Server 2022. * Click on "**Add roles and features**" in the dashboard or the top menu. * In the "Before you begin" page, review the information, and click "**Next**." * Select "**Role-based or feature-based installation**" and click "**Next**." * Choose the local server on which you want to install the feature, and click "**Next**." * In the "Select server roles" page, leave it as default (no roles selected), and click "**Next**." * In the "**Select features**" page, expand "**File and Storage Services**." * Scroll down and check the box next to "**Windows Server Backup**." * A pop-up window will appear. Click on "**Add Features**" and then click "**Next**." * Review the information about the Windows Server Backup feature and click "**Next**." * On the "**Select features**" page, click "**Next**." * In the "**Confirm installation selections**" page, review your choices and click "**Install**." * Wait for the installation to complete, and then click "**Close**."  1. **Set up OneDrive**:  * Create a OneDrive account if you don't have one already. * Install the OneDrive sync client on the Windows Server 2022 machine. * Sign in to the OneDrive sync client using your OneDrive account credentials. |
| 1. **Prepare Backup Storage**:  * Ensure that you have sufficient storage space available in your OneDrive account to accommodate the backup. |
| 1. **Configure Backup Script:**   Open Notepad or any text editor and create a new script to perform the backup. Below is an example of a PowerShell script for backing up the Active Directory database and system state: |
| # Set backup parameters  $backupLocation = "C:\Backup"  # Local temporary folder to store the backup files  $oneDriveFolder = "OneDrive\AD\_Backups"  # The folder in your OneDrive where you want to store the backups  # Create backup folder if it doesn't exist  New-Item -ItemType Directory -Force -Path $backupLocation  # Perform backup  wbadmin start systemstatebackup -backupTarget:$backupLocation  # Move backup files to OneDrive  Move-Item -Path "$backupLocation\WindowsImageBackup" -Destination "$env:USERPROFILE\$oneDriveFolder" |
| 1. Save the script with a **.ps1** extension, such as **backup.ps1** 2. **Schedule the Backup:**  * Open the Task Scheduler on the Windows Server 2022 machine. * Create a new task and provide a name and description. * In the "Actions" tab, specify the following details:     **Action: Start a program**   * Program/script: **powershell.exe** * Add arguments: **-ExecutionPolicy Bypass -File "C:\Path\To\backup.ps1**" * Configure the desired schedule for the backup task (e.g., daily, weekly).      1. **Test the Backup:**   Manually run the backup script to ensure it executes successfully and verifies the backup files are uploaded to OneDrive.  This approach may require additional consideration and customization to fit your specific environment.  It is recommended to thoroughly test the backup and restore process before relying on it in a production environment. |
| **Restore Steps** |
| **To restore the Active Directory from the backup:**   1. **Install Windows Server 2022:**  * Install a fresh copy of Windows Server 2022 on the target machine.  1. **Restore the Backup Files:**  * Log in to the OneDrive account from the new server and ensure that the backup files are synced and available in the appropriate OneDrive folder.  1. **Restore Active Directory**:  * Open the Server Manager and select "**Local Server**" from the left pane. * In the "**Properties**" section, click on the "**Add roles and features**" link. * Follow the wizard and select the "**Active Directory Domain Services**" role for installation. * After the installation, open the "**Active Directory Domain Services Configuration Wizard**" and choose the option to "**Use advanced mode installation**." * Select the "**Windows Server Backup**" option to restore AD from backup.  Specify the location of the backup files on OneDrive and follow the prompts to complete the restore process. |
| **Encryption Option** |
| To encrypt the backup files for added security, you can use the built-in encryption features of Windows Server Backup and OneDrive. Here's how you can encrypt the files during the backup process:   1. **Enable BitLocker Drive Encryption (Optional):**   If you want to encrypt the entire server's hard drive, you can enable BitLocker Drive Encryption on the server. This will provide full disk encryption and protect all files, including the backup files.   1. **Encrypt the Backup Files:**   Modify the backup script to include encryption for the backup files.  **Here's an example modification to the PowerShell script:** |
| # Set backup parameters  $backupLocation = "C:\Backup"  # Local temporary folder to store the backup files  $oneDriveFolder = "OneDrive\AD\_Backups"  # The folder in your OneDrive where you want to store the backups  $encryptionPassword = "YourEncryptionPassword"  # Password for encrypting the backup files  # Create backup folder if it doesn't exist  New-Item -ItemType Directory -Force -Path $backupLocation  # Perform backup with encryption  wbadmin start systemstatebackup -backupTarget:$backupLocation -encryption:$true -encryptionPassword:$encryptionPassword  # Move encrypted backup files to OneDrive  Move-Item -Path "$backupLocation\WindowsImageBackup" -Destination "$env:USERPROFILE\$oneDriveFolder" |
| Save the modified script and use it for the backup task in Task Scheduler.  By specifying the **-encryption:$true** and **-encryptionPassword:$encryptionPassword** parameters in the **wbadmin** command, the backup files will be encrypted using the provided password during the backup process.  Additionally, when using OneDrive for cloud storage, it provides its own encryption mechanisms to secure data during transmission and storage. Your files are encrypted in transit using SSL/TLS protocols and are encrypted at rest using OneDrive's encryption technologies. |

|  |
| --- |
| Linux Web Server |
| Prerequisites: Installing OneDrive |
| Before the backup to OneDrive can be setup as a scheduled task, OneDrive needs to be installed onto the Linux 2 Web Server. Follow the steps below to install OneDrive  **INSTALLING ONEDRIVE**  **Step 1: Update System Packages**  Open up your Ubuntu 22.04 terminal by pressing “**CTRL+ALT+T**” and update the system packages:  $ sudo apt update  **Step 2: Add Release Key of OpenSuSE Build Service Repository**  In the next step, add the release key of the OpenSuSE release key to your system:  $ wget -qO - https://download.opensuse.org/repositories/home:/npreining:/debian-ubuntu-onedrive/xUbuntu\_22.04/Release.key | gpg --dearmor | sudo tee /usr/share/keyrings/obs-onedrive.gpg > /dev/null    **Step 3: Add OpenSuSE Build Service Repository**  After adding the release key of the OpenSuSE build release key, add its repository with the help of the following command:  $ echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/obs-onedrive.gpg] https://download.opensuse.org/repositories/home:/npreining:/debian-ubuntu-onedrive/xUbuntu\_22.04/ ./" | sudo tee /etc/apt/sources.list.d/onedrive.list    **Step 4: Update System Packages**  Update the system packages:  $ sudo apt update  **Step 5: Install OneDrive on Ubuntu 22.04**  Now, install OneDrive on your Ubuntu 22.04 system:  $ sudo apt install onedrive -y    The given error-free output states that we have successfully installed OneDrive:  **Let’s head toward launching OneDrive on the Ubuntu system.**  In order to launch OneDrive, type “onedrive” on the terminal and hit “Enter”:  $ onedrive  Copy the link in the output and paste it into your favourite browser for authorising the OneDrive application by utilizing your account:    Permit the OneDrive client the required access and click on the “**Accept**” button:  After doing so, you will be navigated toward the following blank page. Copy the URI from the address bar:    Then, paste the copied URI in front of the “Enter the response uri” section and press “Enter”:    Upon doing so, you will see a message signifying that the OneDrive application is successfully authorized:  Synchronize your cloud OneDrive file to the local system by specifying the “–**synchronize**” option in the “**onedrive**” command:  $ onedrive --synchronize    Depending on the amount of files, this process can take a while to complete.  **The following section will detail how to use OneDrive with the help of a GUI tool**  “**insync**” is a client utility that can be used to manage your oneDrive, Google Drive, and Dropbox files on Ubuntu desktop.  It can sync any files and folders from the local storage in a few seconds.  To sync files on OneDrive using insync, follow the below-given procedure.  **Step 1: Install insync**    Firstly, visit the official website of the insync, click on the “For desktop” button, and select the “Download for Linux” option: <https://www.insynchq.com/downloads/linux>  **Step 2: Select Distribution and Version**  From the highlighted drop-down menu, select “Ubuntu” as your distribution and its release version:  Within a few minutes, insync “.deb” file will start downloading:  **Step 3: Move to Downloads folder**  Move to “Downloads” directory where the insync “.deb” file is placed:  $ cd Downloads/  **Step 4: Install insync**  Run the following command to install insync on your system:    $ sudo apt install ./insync\*.deb  You can also utilize the Ubuntu 22.04 GUI for the specified purpose. To do so, move to the “**Downloads**” directory, right-click on the insync “.deb” file, and select the “**Open With Other Application**” option:  Choose the “**Software Install**” option and click on the “**Select**” button  Then, click on the “**Install**” button  You will be then asked to enter the system password for the authentication    As you can see, we have successfully installed insync on our system:    **Step 5: Launch insync**  Search for “**insync**” in the “**Applications**” menu of Ubuntu and launch it:  Click on the “Start Insync” button to start using the insync application:  **Step 6: Connect insync with OneDrive**    Hit the “Add OneDrive” option to sign into OneDrive via browser:  Enter your “Email Address” linked with Microsoft account and click on the “Next” button:  Authorize required permissions to insync and click on the “**Accept**” button:    **Step 7: Setup Account**  Click on the “Next” button to setup your account on insync:  Then, select the base folder or the local default sync folder and hit “Next”:  Configure your sync preferences and click “Next”:  **Step 8: Sync Files on OneDrive**  Choose the items you want to 2-way sync from files and click on the “**Sync**” button.    The green verification mark with the highlighted files indicates that we have successfully synced our files to OneDrive   |  | | --- | | Configure Sophos Automatic Backups |   **Setting up the Sophos Firewall to perform scheduled backup settings.**  In order for the backup to successfully save the current settings of the Sophos firewall, we need to configure the firewall to perform regular backups of its configuration files, that we can then access vie SSH and stored on the cloud.     1. Log into the Sophos GUI using a browser, using the address of the server, followed by the port number 4444   E.g. **192.168.2.100:4444**   1. Enter the your credentials and log into the firewall.      1. On the left hand navigation menu,  click on **‘Backup & Firmware’**. 2. In the main windows, navigate to the **‘Backup & Restore’** tab, and you will see the details required to make backups to the server. 3. Adjust the settings as shown: Backup mode: **Local** Backup Prefix: (**optional**) Frequency: **Daily** Schedule : **06:00 hrs** Encryption password: (**optional**) 4. Press ‘Apply’ to save the settings, then click ‘Backup now’ to run a new backup that we can use to store on the OneDrive. |
| Backup Steps |
| To perform this backup, a Bash script was created to automate the backup to the OneDrive |
|
| 1. #!/bin/bash 2. # Set the OneDrive backup location directory 3. ONEDRIVE\_DIR="/home/ozcazual/Insync/cyberclass2023@outlook.com/OneDrive/Linux Backup" 4. # Set the backup filename with timestamp 5. BACKUP\_FILENAME="backup\_$(date +%Y%m%d%H%M%S).tar.gz" 6. # Connect to Sophos firewall and backup configuration files 7. ssh "$SOPHOS\_USERNAME@$SOPHOS\_IP" "scp -r /var/conf/backupdata ozcazual@192.168.2.101:~/server\_backup/sophos\_firewall" 8. # Set the directories and files to include in the backup 9. BACKUP\_SOURCES=( 10. "/var/www" # OzCazual e-commerce website 11. "/home/ozcazual/server\_backup/sophos\_firewall" # Firewall configuration 12. "/opt/splunk" # Splunk directory 13. "/opt/splunkforwarder" # Splunkforwarder directory 14. "/etc/snort" # Snort directory 15. "/etc/wireshark" # Wireshark directory 16. "/opt/sophos-spl/plugins/av" # Sophos Intercept X directory 17. ) 18. # Perform the backup using tar with root privileges 19. sudo tar -czvf "$BACKUP\_DIR/$BACKUP\_FILENAME" "${BACKUP\_SOURCES[@]}" >/dev/null 2>&1 20. # Check if insync is running, and if not, start it 21. INSYNC\_PID=$(pgrep insync) 22. if [[ -z "$INSYNC\_PID" ]]; then 23. insync start 24. fi 25. # Wait for insync to synchronize the files 26. sleep 10 27. # Move the backup file to the OneDrive directory with root privileges 28. sudo mv "$BACKUP\_DIR/$BACKUP\_FILENAME" "$ONEDRIVE\_DIR" 29. # Display a success message 30. echo "Backup created and uploaded successfully." |
| The script performs the following functions:   1. Firstly, we set some global variables, such as the backup directory, the Sophos Firewall details, and the location of the directory the backup will be stored on the OneDrive cloud. 2. Once all the files are added to an archive, the filename suffix is defined, so that multiple backups can be kept. 3. The next section is an important one, as this connects to the Sophos firewall using OpenSSH, and downloads the latest backup of the firewall configuration. 4. The script then locates the configuration files or programs we wish to install. In this case, we are backing up the Apache2 Server, Splunk, Snort, Sophos, Wireshark and the Splunk forwarder. 5. Because we are accessing folders in the server that require elevated access, we include the ‘**sudo**’ command to assist in accessing these areas, without an error. 6. Just in case the insync app isnt running, we check this, and if it isnt, we can start the synchronzation. 7. We give insync 10 secs to sync, before we start the backup process 8. Once all the files are added to the compressed archive, it now moves the file to the OneDrive directory 9. Once all this is complete, the message “Backup created and uploaded successfully” is display.      |  | | --- | | Creating a scheduled automatic backup: |   With the script tested and ready to run when we choose, we can now setup a regular schedule to perform automatic backups.  Typing the command **‘contrab -e’** will start the process.  If its the first time running the command, you will be prompted to select an editor.  The default is nano, and you may press **‘Enter’** or **‘1’** to continue    A file will open, and we can enter a command at the bottom to schedule our backup.  As explained, the format of the schedule is defined.  In this particular case, i have set the backup to start at 6:00 am everyday, specifying what file to run.    Press “**CTRL + X**” to save and exit the file.  The automatic schedule has been created. |
| Restore Steps |
| To restore the saved backup, including the web server, configuration files for Splunk, Splunk Forwarder, Snort, Wireshark, Sophos firewall configuration, and Sophos Intercept X config, from the compressed .tar file stored in OneDrive, you can follow these steps:   1. **Prepare the Linux system**:  * Install a fresh instance of the Linux operating system on the server where the web server and security programs were previously installed. * Ensure that the necessary dependencies and prerequisites for the web server, Splunk, Snort, Wireshark, and Sophos Intercept X are installed.  1. **Download the backup file from OneDrive**:  * Access your OneDrive account and navigate to the folder where the backup file is stored. * Download the compressed .tar backup file to the Linux system. You can use the rclone command-line tool or the OneDrive web interface to download the file.  1. **Extract the backup archive**:  * Open a terminal or SSH session on the Linux system. * Navigate to the directory where the backup file is downloaded. * Use the following command to extract the backup archive:   tar -xzvf backup\_file.tar.gz -C /path/to/destination   * Replace **backup\_file.tar.gz** with the actual filename of the downloaded backup file. * Replace **/path/to/destination** with the path to the directory where you want to extract the backup.   This should be the root directory ("/") if you want to restore the web server and configuration files to their original locations.   1. **Restore the web server:**  * Depending on the specific web server you were using (such as Apache or Nginx), you may need to copy the extracted web server files to the appropriate directories. * Identify the directories where the web server files were previously stored and copy the extracted web server files to those locations.  1. **Restore configuration files:**  * Locate the extracted configuration files for Splunk, Splunk Forwarder, Snort, Wireshark, Sophos firewall, and Sophos Intercept X. * Copy the respective configuration files from the extracted backup to their original locations on the Linux system.  1. **Start the services:**  * Start the web server and other security programs (Splunk, Snort, Wireshark, Sophos Intercept X) using their respective commands or services. * Refer to the documentation or instructions for each program to ensure they are properly started and configured.  1. **Verify the restoration:**  * Test the web server by accessing the website and ensure that it is functioning as expected. * Verify the functionality and configuration of the restored security programs (Splunk, Snort, Wireshark, Sophos Intercept X).   By following these steps, you should be able to restore the web server and configuration files for Splunk, Splunk Forwarder, Snort, Wireshark, Sophos firewall, and Sophos Intercept X from the compressed .tar backup file stored on OneDrive. Remember to review the specific instructions and configurations for each program to ensure a successful restoration. |

|  |
| --- |
| **Backup Locations** |
| The location of each programs installation directory and configuration files can vary, depending on if the installation was customised in any way during. For the purpose of this document, all software was installed in the default installation locations specified during the installation.  If the installation of an additional plugins was performed, it may also be required to locate the original installation files and the installation directory where configuration files may be stored.  These are the default locations for the configuration files of each of the security software programs installed on the Windows 2022 AD Server.   |  |  |  | | --- | --- | --- | |  | **Configuration files** | **Additional data** | | **pfSense firewall** | Typically stored in the /conf directory within the pfSense installation directory. | Firewall rules, NAT configurations, and other settings are stored in the pfSense configuration files | | **Splunk** | Found in the Splunk configuration directory. The default path is C:\Program Files\Splunk\var\lib\splunk | Located in the Splunk data directory.  The default path is  C:\Program Files\Splunk\etc | | **Snort** | Typically located in the Snort installation directory. The default path is C:\Snort\etc | Stored in the rules subdirectory within the Snort installation directory | | **Wireshark** | By default, the user's profile directory is C:\Users\username\AppData\Local\Temp\Wireshark. | Wireshark's preferences and configuration files are saved in the user's profile directory. The path is typically C:\Users\username\AppData\Roaming\Wireshark | | **Sophos Antivirus** | Located in the Sophos antivirus installation directory.  The default path is C:\Program Files\Sophos | Stored in various locations within the Sophos antivirus installation directory,  such as C:\Program Files\Sophos\Logs | |
| **Backup Steps** |
| * + - 1. Before the backup can be setup, some files are required to automate the backup process. Use the following address to download automated tools that can be downloaded from GitHub at the following addresses:   [https://github.com/badbread/breadsPFsenseAutoBackup](https://github.com/badbread/breadsPFsenseAutoBackup%20)   1. Click on the ‘Code’ button, and click on the ‘Download ZIP’ button. 2. A download will begin, and a file called “breadsPFsenseAutoBackup-master” 3. 3. Unzip the archive and install it into a root directory, such as “C:\pfSense” |
| 1. Go to the URL <https://github.com/KoenZomers/pfSenseBackup/releases/tag/2.6.0> 2. Download the file named ‘**pfSenseBackupv2.6.0-win-x64.exe**’  1. Move the .**exe** file into the directory where the archive in the last step was extracted to. 2. Rename the .**exe** file to ‘**pfSenseBackup.exe**’ 3. While in the directory, locate the file ‘**backupPFsense.ps1’** right-click on the file, and open it with a text editor such as Notepad.   When the file open, there are markers for where code needs to be tailored to our backup. Below is a small section of the code, configured for the backup. |
| $backupdir = "C:\Users\Administrator\OneDrive\Backup\pfSense\" #where your backup files will go (leave a trailing backslash)  $appdir = "C:\pfSense\" #where the exe lives (leave a trailing backslash)  $username = "backup" #PFSense Username  $pw = "pfsense" #PFSense Password  $pfaddress = "192.168.1.1" #PFSense address  $retention = "-30" #how many days to keep old backup files (Must be negative value, -30 = files older than 30 days)  $fileName = "pfsensebackup" #name of your log file (put into your $backupdir)  $minbackups = "5" #minimum number of backups to keep |
| There is code that needs to be inserted to include the backup of the programs listed above.  The code locates the configuration files of each program, copies and stores a backup on the OneDrive directory. The highlighted area is the code for the program backup. |
| # Push options  $usepush = "n" #use pushover? (y or n)  $pushalways = "y" #send push on good backups  $pushoverapp = "c:\somepath\pushovercli.exe" #location of the pushovercli  $pushsubject = "pfSense Backup" #subject of the push message  #Should not need to be modified  $logFileName = $fileName + ".log"  $log = $backupdir + "\" + $logFileName  $HowOld = (Get-Date).AddDays($retention);  $command= $appdir + "pfSenseBackup.exe" + " -u " + $username + " -p " + $pw + " -s " + $pfaddress + " -o " + $backupdir + " -usessl "  function write-log {  param(  [parameter(Mandatory=$true)]  [string]$text,  [parameter(Mandatory=$true)]  [ValidateSet("WARNING","ERROR","INFO", "AppOutput")]  [string]$type  )  [string]$logMessage = [System.String]::Format("[$(Get-Date)] -"),$type, $text  Add-Content -Path $log -Value $logMessage  }  function create\_backup {  write-log -text "Executing the backup command" -type INFO  $cmdOutput = Invoke-Expression $command | Out-String  write-log -text $cmdOutput -type AppOutput  if ($cmdOutput -like '\*DONE\*' )  {  write-log -text "Copy completed successfully." -type INFO  deloldbackups;  }  else {  write-log -text "ERROR running copy exe" -type ERROR  sendpush -message "PFSense auto backup failed!"  exit  }    }  function deloldbackups {  $backupcount = ( Get-ChildItem $backupdir -filter \*.xml | Measure-Object ).Count  if ($backupcount -gt $minbackups) {  write-log -text "Attempting to delete old backups..." -type INFO;  get-childitem -Path $backupdir -recurse | where-object { !$\_.PSIsContainer -and $\_.lastwritetime -lt $HowOld } | remove-item -force  write-log -text "Backups older than $HowOld days deleted." -type INFO;  alldone;  }  else {  write-log -text "There are not at least $minbackups backup files, not deleting anything..." -type WARNING;  alldone;  }  }  function alldone {  write-log -text "Backup completed successfully" -type INFO;  write-host "Backup complete."  if ($pushalways = "y") {  sendpush -message "PFSense auto backup completed successfully"  exit  }  else {  exit  }  }  # Set the backup parameters for each program  $Programs = @{  "Splunk" = "C:\Program Files\Splunk\etc";  "Snort" = "C:\Snort";  "Wireshark" = "C:\Users\Administrator\AppData\Roaming\Wireshark";  "Sophos Firewall" = "C:\Program Files\Sophos"  }  $OneDriveFolderPath = "C:\Users\Administrator\OneDrive\backup\ProgramBackups"  # Create backup folder  $BackupFolder = Join-Path -Path $OneDriveFolderPath -ChildPath "ProgramConfigBackup\_$(Get-Date -Format 'yyyyMMdd')"  New-Item -ItemType Directory -Path $BackupFolder -Force | Out-Null  # Backup program configurations  foreach ($program in $Programs.GetEnumerator()) {  $ProgramName = $program.Key  $ProgramPath = $program.Value  $ProgramBackupFolder = Join-Path -Path $BackupFolder -ChildPath $ProgramName  if (Test-Path -Path $ProgramPath) {  Copy-Item -Path $ProgramPath -Destination $ProgramBackupFolder -Recurse -Force  Write-Host "Backup of $ProgramName configuration completed successfully."  } else {  Write-Host "Error: $ProgramName not found at $ProgramPath."  }  }  # Set the backup parameters  $Programs = @{  "Splunk" = "C:\Program Files\Splunk\etc";  "Snort" = "C:\Snort\etc"; “C:\Snort\rules”;  "Wireshark" = "C:\Users\Administrator\AppData\Roaming\Wireshark";  "Sophos Firewall" = "C:\Program Files\Sophos"  }  $OneDriveFolderPath = "C:\Users\Administrator\OneDrive\ProgramBackups" #  # Create backup folder  $BackupFolder = Join-Path -Path $OneDriveFolderPath -ChildPath "ProgramConfigBackup\_$(Get-Date -Format 'yyyyMMdd')"  New-Item -ItemType Directory -Path $BackupFolder -Force | Out-Null  # Backup program configurations  foreach ($program in $Programs.GetEnumerator()) {  $ProgramName = $program.Key  $ProgramPath = $program.Value  $ProgramBackupFolder = Join-Path -Path $BackupFolder -ChildPath $ProgramName  if (Test-Path -Path $ProgramPath) {  Copy-Item -Path $ProgramPath -Destination $ProgramBackupFolder -Recurse -Force  Write-Host "Backup of $ProgramName configuration completed successfully."  } else {  Write-Host "Error: $ProgramName not found at $ProgramPath."  }  }  function sendpush($message) {  if ($usepush = "y") {  write-log -text "Push notification sent" -type INFO;  $message = $message.ForEach({ "$pushsubject - " + $message})  & $pushoverapp message="$message";  exit  }  else {  write-log -text "Push not used or variable incorrect"  exit  }  }  Add-Content -Path $log -Value "--------------------------------------------------------------------"  Add-Content -Path $log -Value "Backup process started"  Create\_backup; |
| 1. The full script is now ready to save.  Save the script, ensure the file extension remains a “**.ps1**” format. This is important, as this file will be used now to insert a task into the Task Scheduler. 2. Click the Windows ‘**Start**’ button, and begin typing **‘Task Manager**’. 3. When the Task Manager loads, from the menu bar, click on ‘**Action > Create Task…**’ |
| 1. In the new task window, create a name for the task. Then click on the ‘**Triggers Tab**’  1. Choose the frequency of the backup, then press ‘**OK**’ 2. Click on the ‘**Actions**’ tab, then click ‘**New….**’ 3. In the ‘**Action**’ pull down menu, select ‘**Start a program**’ 4. In the ‘**program / script:**’ field, enter ‘**powershell.exe**’ 5. In the ‘Add arguments’ section, include the following: **-ExecutionPolicy Bypass -File "C:\Path\To\backup.ps1**" Click ‘**OK**’ when done.   The task has now been created and scheduled.  This can be tested, by right-clicking on the newly created task and click **‘Run’** |
| To perform a restoration of the saved backup, including the configuration files for Splunk, Splunk Forwarder, Snort, Wireshark, pfSense firewall, and Sophos antivirus, you can follow these detailed steps:  **Prepare the server**:   * Install a fresh instance of Windows Server 2022 on the server where the AD server and security programs were previously installed. * Ensure that the necessary prerequisites and dependencies for Splunk, Splunk Forwarder, Snort, Wireshark, pfSense firewall, and Sophos antivirus are met.   **Download the backup files from OneDrive:**   * Access your OneDrive account and navigate to the folder where the backup files are stored. * Download each folder corresponding to the program name (e.g., Splunk, Splunk Forwarder, Snort, Wireshark, pfSense firewall, Sophos antivirus) to the appropriate location on the server.   **Restore configuration files:**  For each program (Splunk, Splunk Forwarder, Snort, Wireshark, pfSense firewall, Sophos antivirus):   * Locate the respective folder that contains the configuration files within the downloaded backup files. * Copy the configuration files from each program's folder to their original locations on the server. * Refer to the documentation or instructions for each program to determine the specific locations where the configuration files need to be restored.   **Install and configure software:**   * Install each program (Splunk, Splunk Forwarder, Snort, Wireshark, pfSense firewall, Sophos antivirus) on the server if they are not already installed. * Use the appropriate installation files or package managers for each program to install them according to their documentation. * Configure each program based on the restored configuration files and your specific requirements. * Refer to the documentation or instructions for each program to properly configure them based on the restored configuration files.   **Verify the restoration:**   * Start each security program (Splunk, Splunk Forwarder, Snort, Wireshark, pfSense firewall, Sophos antivirus) and verify that they are functioning correctly. * Test the integration between the security programs to ensure they are properly communicating and protecting the network.   By following these steps, you should be able to restore the configuration files for Splunk, Splunk Forwarder, Snort, Wireshark, pfSense firewall, and Sophos antivirus from the backup stored on OneDrive. It's important to review the specific instructions and configurations for each program to ensure a successful restoration. |
| **Restore Steps** |