

Backing Up and Restoring Data in Windows and Linux

Scenario

Data availability is a key component of security. Hard disk drives fail, data is corrupted or accidentally deleted, or changed in an unplanned manner. In this activity, you will back up data on a Windows server and a Linux Server. You'll store the backup jobs on remote servers, which helps protect the data in case of drive failure on the original server. You'll delete the backed up data to simulate its loss, and then restore it from the remote servers to a different location on the original servers.

Objectives

This activity is designed to test your understanding of and ability to apply content examples in the following CompTIA Security+ objective:

- 2.5 Given a scenario, implement cybersecurity resilience.

Lab

- DC1 VM
- MS1 VM
- Kali VM
- LX1 VM
- pfSense VM

Task 1

Backup files on Windows

You will backup files on the Windows server and store the backup job on a remote server.

NOTE: To improve the lab performance turn on only **DC1**, **MS1**, and **pfSense** VMs.

1. On the **DC1**, send **CTRL+ALT+DEL**, and then sign on using **CONTOSO\Administrator** and **Pa\$\$w0rd** as the password.
2. Open **File Explorer**, browse to the **C:** drive, and then create a new folder named **Backups**.
3. To share the **C:\Backups** folder, right-click it, select **Properties**, select the **Sharing** tab, select **Advanced Sharing**, and then check the box for **Share this folder**.

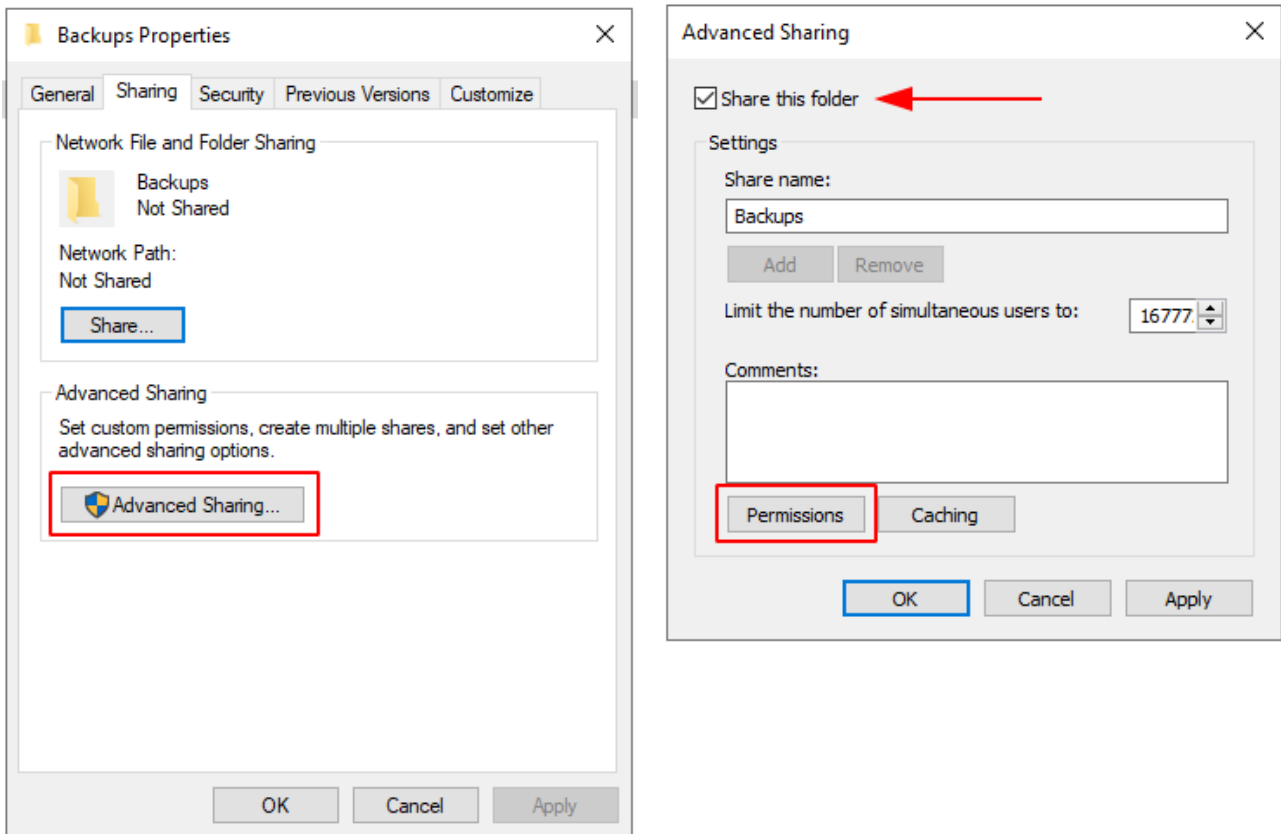


Figure 1.1 – Sharing a Folder in Windows.

4. Select the **Permissions** button. Share the folder with the **Everyone** group at **Full Control**.

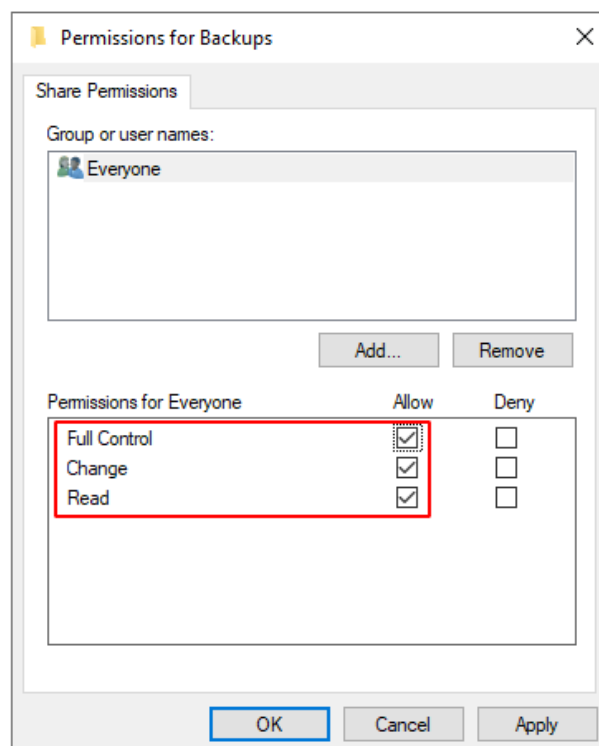


Figure 1.2 – Assigning the Everyone group Full Control.

NOTE: The folder is only being left open for the purposes of this activity. You would normally apply NTFS permissions to restrict access to a backup operator account.

5. Select **Apply** and then confirm each dialog box.

6. Switch to the **MS1** VM and send **CTRL+ALT+DEL**. Sign in by using **CONTOSO\Administrator** and **Pa\$\$w0rd**.

7. From **Server Manager**, select **Tools > Windows Server Backup**.

Most backup operations would run to a schedule, but for this activity you will configure a manual backup on the **MS1** server.

8. From the **wbadmin** console, select the **Local Backup** node on the left pane.

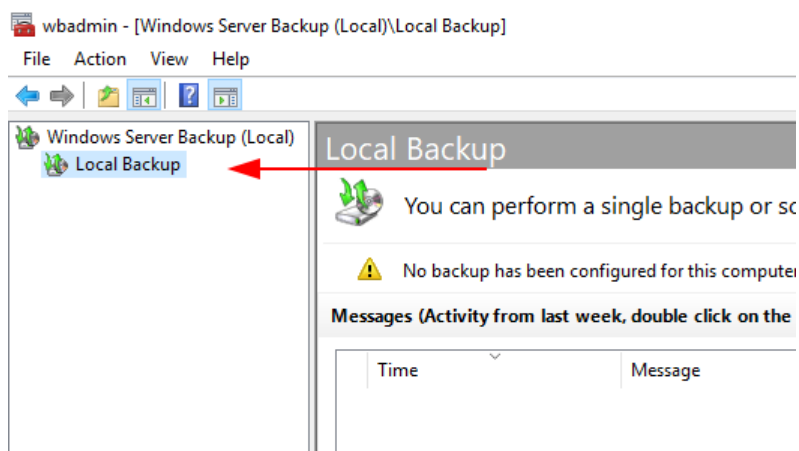


Figure 1.3 – wbadmin Local Backup node.

It takes about one minute for the backup program to check the server's configuration.

9. From the **Actions** pane on the right, select **Backup Once**, then select **Next**.

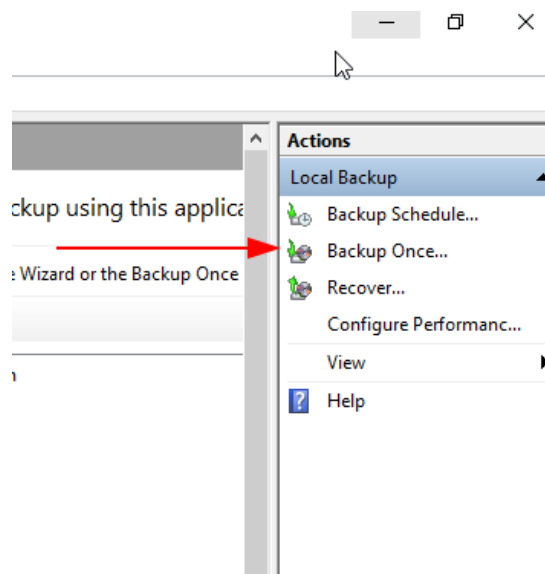


Figure 1.4 – wbadmin Actions pane.

10. Select **Custom**, then select **Next**.

11. Select **Add Items** and browse to drive **C:\sources** and then check the box next to the **sxs** folder to backup the entire folder. Select **OK**, and then select **Next**.

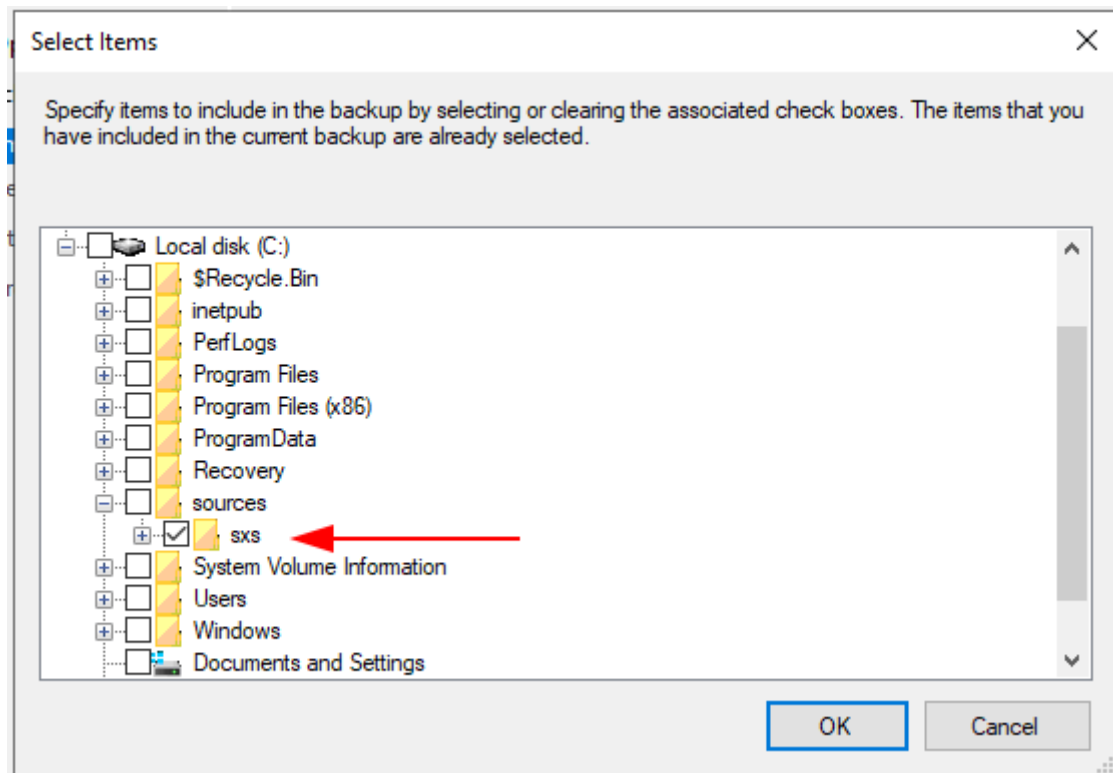


Figure 1.5 Selecting the sxs folder to back up.

12. Select **Remote Shared Folder**, and then **Next**.

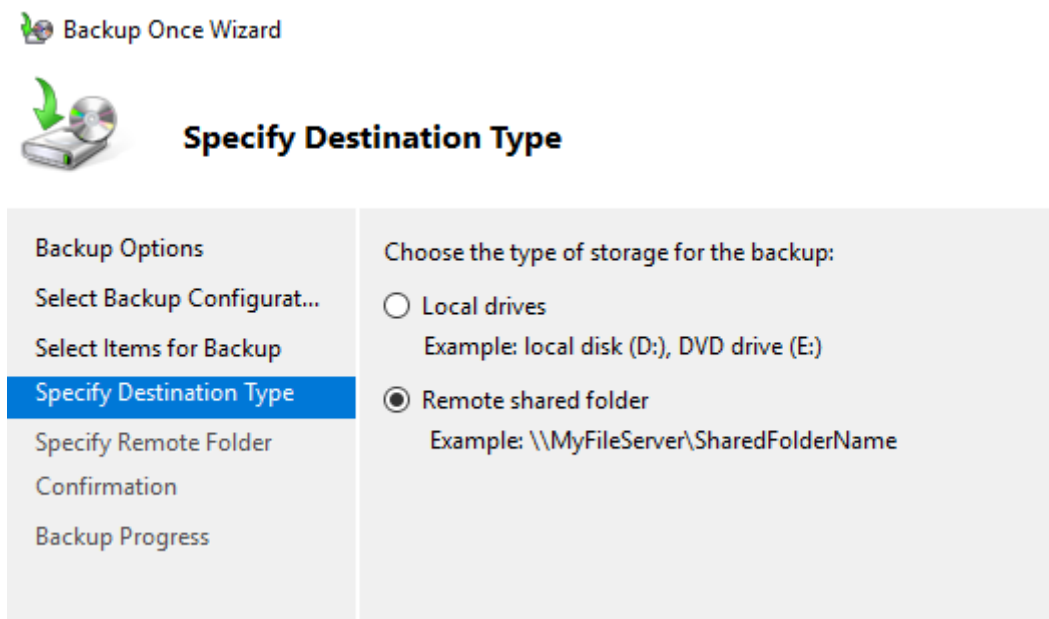


Figure 1.6 – Specifying a Destination Type of Remote shared folder.

13. Enter **\\DC1\Backups**, select **Next**, and then select **Backup**.

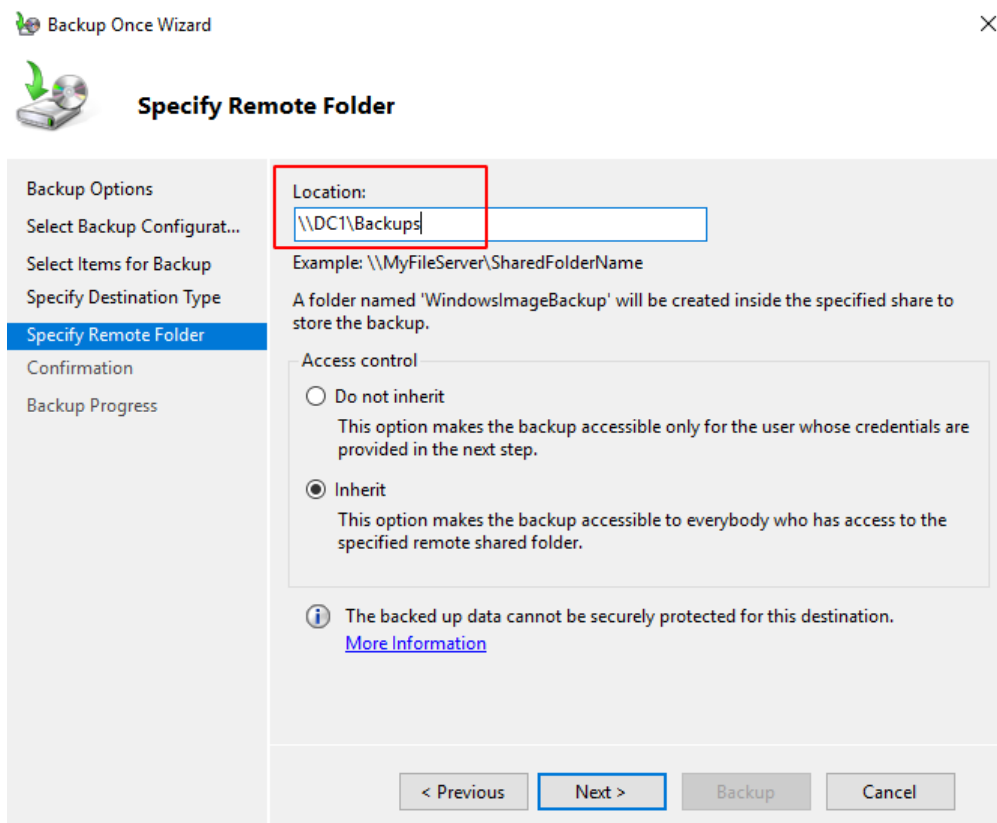


Figure 1.7 – Specify a Remote Folder location.

The backup job takes about two minutes to complete.

14. Confirm the contents of the C:\Backups folder on DC1 contains the WindowsImageBackup file.

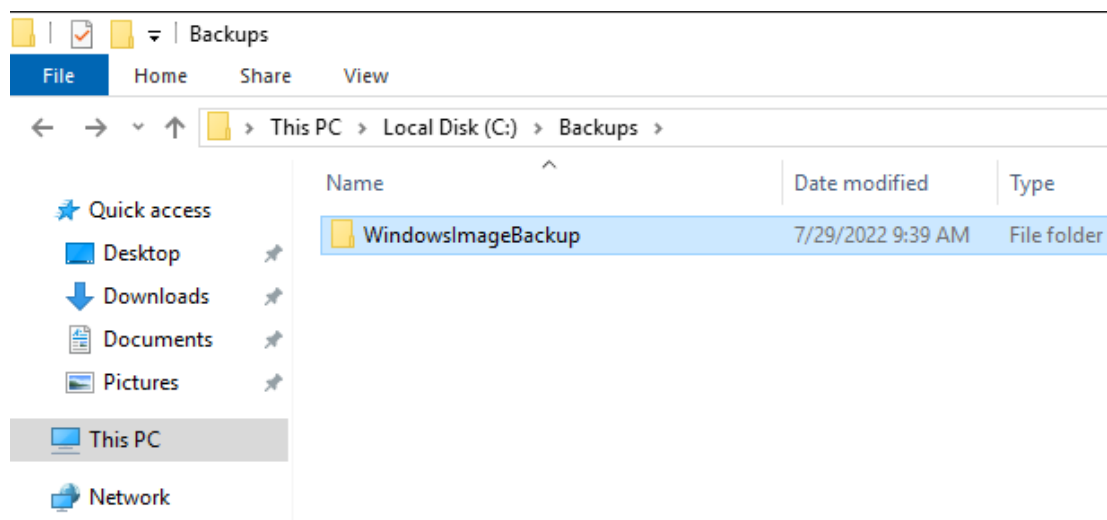


Figure 1.8 – The WindowsImageBackup folder containing the MS1 backup files.

15. In the Backup Once Wizard, select **Close**. Leave the wbadmin console open.

16. Delete the **sxs** folder **contents** from the **MS1** server to simulate the loss or corruption of data.

Task 2

Restore files on Windows

In this activity, you will restore the files you “accidentally” deleted in the previous task.

1. On the **MS1** VM, in the **wbadmin** console, select **Recover** from the **Actions** menu on the left to begin the restore process.

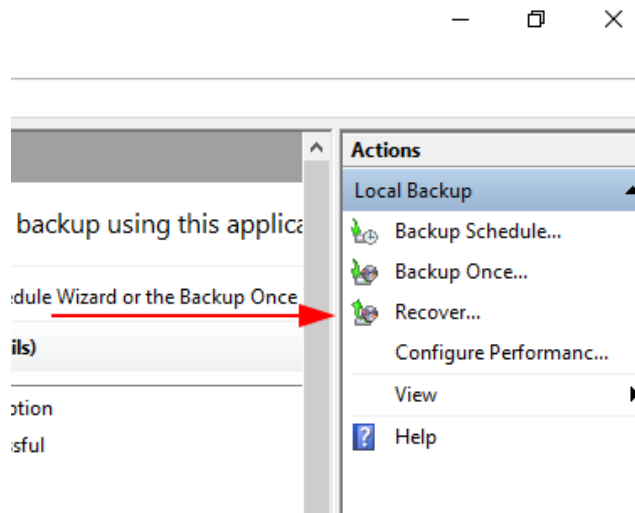


Figure 2.1 – wbadmin console Recover Action.

2. Select **A backup stored on another location**, and then select **Next**.
3. Select **Remote shared folder** and then select **Next**.
4. Enter **\\DC1\Backups**, and then select **Next**.

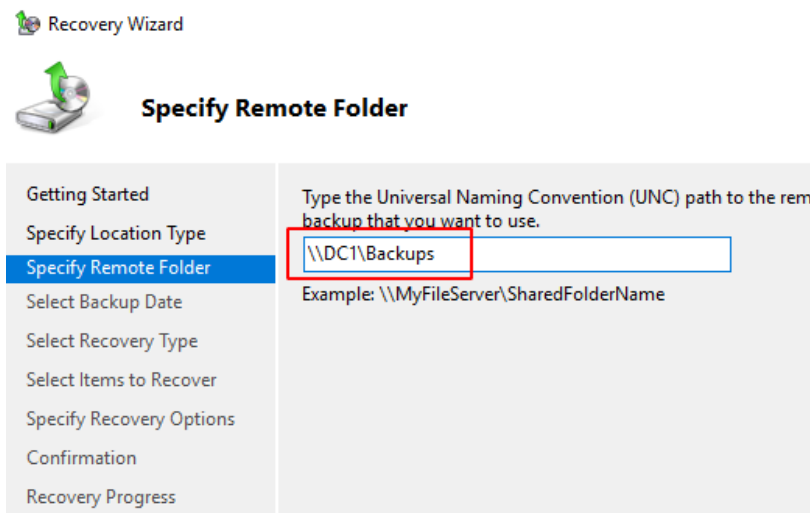


Figure 2.2 – Recovery wizard specifying a remote folder location.

5. Ensure the current date is selected, and then select **Next**.
6. Choose to restore **Files and folders**, and then select **Next**.

7. In the **Select items to Recover** window, browse to the **MS1 > Local disk (C:) > sources > sxs**.

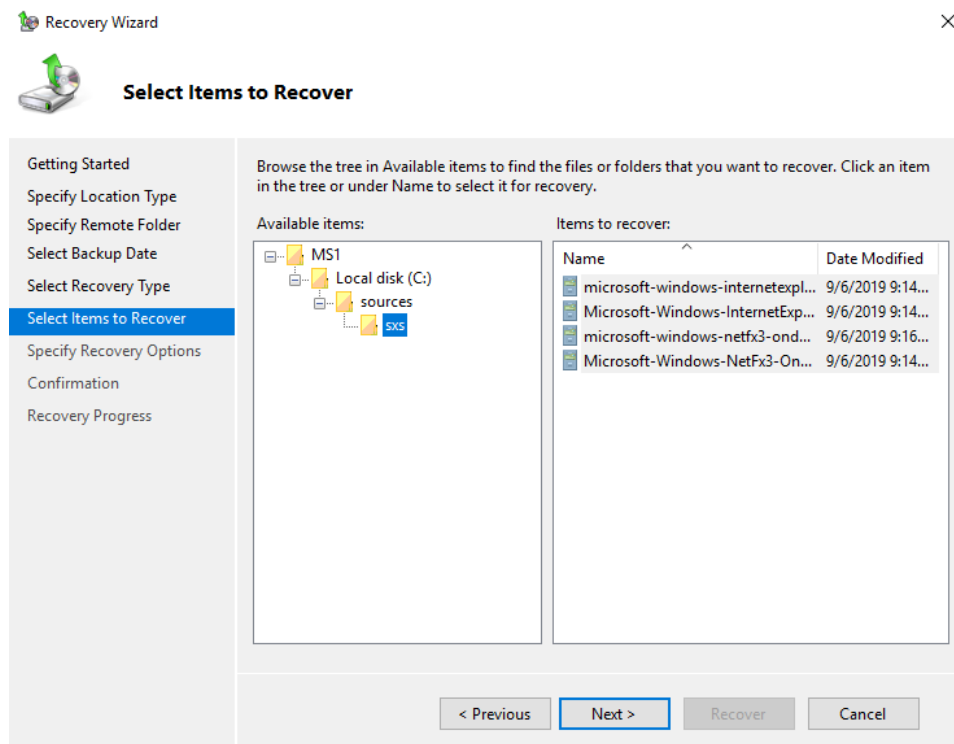


Figure 2.3 – Selecting the items to recover.

You will recover the entire folder, but you can also restore individual files.

8. In the **Specify Recovery Options** window, select the **Browse** button, and then select **This PC > Local Disk (C:)**, and select **sources > sxs**, and select **OK**.

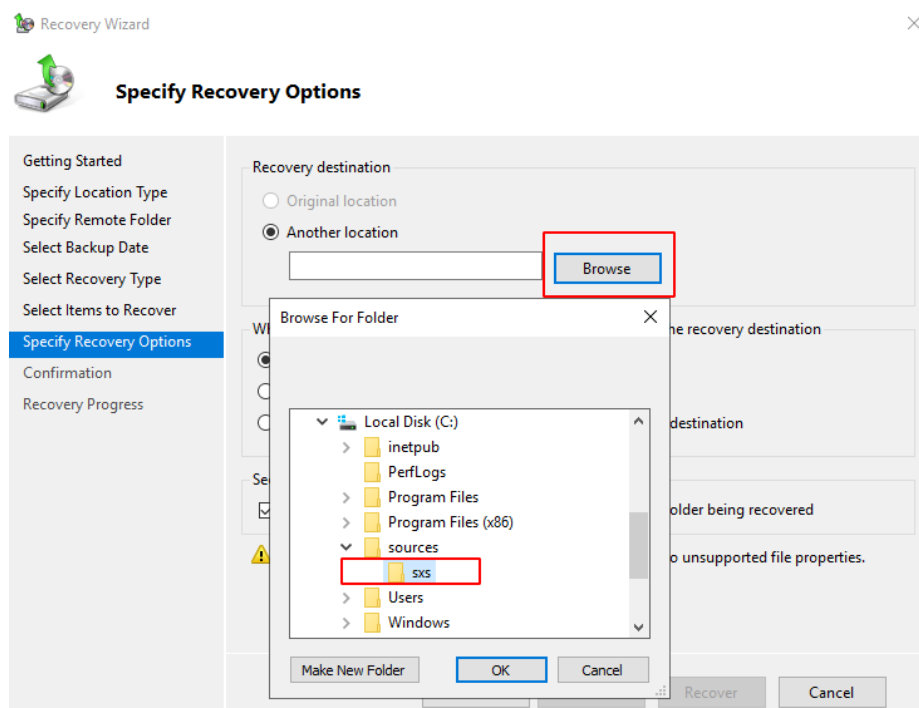


Figure 2.4 – Selecting a folder to restore data.

9. In the **Specify Recovery Options** window, select **Next**.

10. Select **Recover**.

The recovery process takes about one minute.

12. Close the **Windows Server Backup** utility.

13. Browse to **C:\sources\sxs** and display the restored files to confirm the recovery process worked correctly.

NOTE: To improve the lab performance you can turn off **DC1** and **MS1** VMs and turn on **Kali** and **LX1** VMs.

Task 3

Backup files on Linux

In this activity, you will create a few resources, and then perform a basic backup by using the **tar** command. You will store the backup job on a remote Linux server. Finally, you will delete the resources to simulate lost or corrupted data.

1. Sign in to the **Kali** VM as **kali** using **Pa\$\$w0rd**.

2. From the top bar, open the **Terminal Emulator** icon.

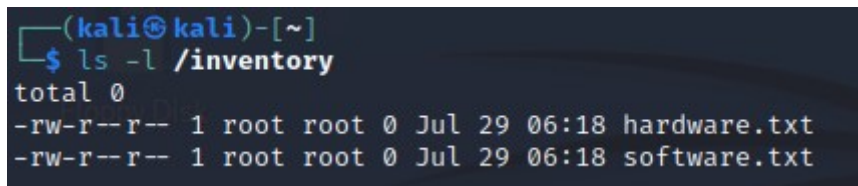
3. Run the following commands to create a directory and several files:

```
sudo mkdir /inventory
```

```
sudo touch /inventory/hardware.txt /inventory/software.txt
```

4. Run the following command to display the contents of the **/inventory** directory:

```
ls -l /inventory
```



```
(kali㉿kali)-[~]  
$ ls -l /inventory  
total 0  
-rw-r--r-- 1 root root 0 Jul 29 06:18 hardware.txt  
-rw-r--r-- 1 root root 0 Jul 29 06:18 software.txt
```

Figure 3.1 – Listing the contents of the /inventory folder.

5. Enter **cd /inventory** to change to the inventory directory.

6. Run the following command to create a backup job named **inventory.tar**:

```
sudo tar -cf inventory.tar /inventory
```



```
(kali㉿kali)-[/inventory]
$ sudo tar -cf inventory.tar /inventory
tar: Removing leading '/' from member names
tar: /inventory/inventory.tar: file is the archive; not dumped
```

Figure 3.2 – Creating a tarball file from the /inventory directory.

7. Run the following command to display the contents of the inventory tarball:

```
sudo tar -tf inventory.tar
```

```
(kali㉿kali)-[/inventory]
$ sudo tar -tf inventory.tar
inventory/
inventory/software.txt
inventory/hardware.txt
```

Figure 3.3 – Listing the contents of the inventory.tar tarball.

8. Create a new file named **devices.txt** in the /inventory directory:

```
sudo touch /inventory/devices.txt
```

9. Append the new **devices.txt** to the existing **inventory.tar** tarball:

```
sudo tar -rf inventory.tar /inventory/devices.txt
```

```
(kali㉿kali)-[/inventory]
$ sudo tar -rf inventory.tar /inventory/devices.txt
tar: Removing leading '/' from member names
tar: Removing leading '/' from hard link targets
```

Figure 3.4 – Adding devices.txt file to the tarball.

10. Display the contents of the **inventory.tar** tarball again using the **-tf** option, as you did a few steps above to confirm that the **devices.txt** file has been added.

```
(kali㉿kali)-[/inventory]
$ sudo tar -tf inventory.tar
inventory/
inventory/software.txt
inventory/hardware.txt
inventory/devices.txt
```

Figure 3.5 – Displaying the contents of the inventory.tar file.

11. Run the following command to generate a hash of the **inventory.tar** backup job:

```
sudo md5sum inventory.tar
```

```
(kali㉿kali)-[/inventory]
$ sudo md5sum inventory.tar
534314a31037ac3bfee4bfb6b46dfd75  inventory.tar
```

Figure 3.6 – Generating a md5 hash of inventory.tar.

The hash is displayed on the screen. This value uniquely identifies the state of the file. Any changes to the file, such as corruption during network transfer, would cause the md5sum hash command to generate a different hash value. In this step, you will generate a second hash and confirm that they are identical. Hashing helps to guarantee file integrity.

12. Run the md5sum hash command a second time. And redirect the results into a file named backuphash in your home directory.

```
md5sum inventory.tar > /home/kali/backuphash
```

13. Run the following command to copy the backup job to the **LX1** VM by using **scp**:

```
scp inventory.tar user@192.168.1.8:/home/user/inventory.tar
```

14. Enter **yes** and **Pa\$\$w0rd** when prompted.

```
(kali㉿kali)-[/inventory]
$ scp inventory.tar user@192.168.1.8:/home/user/inventory.tar
user@192.168.1.8's password:
inventory.tar                               100%  10KB  1.8MB/s  00:00
```

Figure 3.7 – Transferring files with scp.

The **scp** command provides a secure copy function across a network. It is easy way of transferring files. You have now stored your backup job on a remote server.

15. Run the following commands to delete the /inventory directory to simulate a mistake or corrupted files.

```
cd
```

```
sudo rm -fR /inventory
```

NOTE: If you type the **cd** command with no arguments, the system automatically returns you to your home directory.

16. Run the **ls -l** to confirm that **/inventory** directory has been deleted. This simulates an accidentally-deleted directory or corrupted files.

Task 4

Restore files on Linux

In this activity, you will restore your “accidentally” deleted files from the remote server.

1. On the **Kali** VM, run the `scp` command to copy the backup job from the **LX1** VM.

```
scp user@192.168.1.8:/home/user/inventory.tar /home/kali/
```

2. Enter **Pa\$\$w0rd** when prompted.



```
(kali㉿kali)-[~]  
$ scp user@192.168.1.8:/home/user/inventory.tar /home/kali/  
user@192.168.1.8's password:  
inventory.tar                               100%  10KB  1.1MB/s   00:00
```

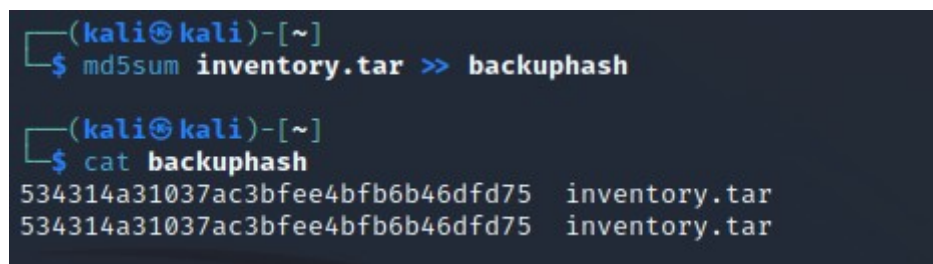
Figure 4.1 – Transferring files from LX1 using scp.

3. Check the **inventory.tar** file on the **Kali** VM by using the **md5sum** command to ensure there was no corruption during transfer. Append it to the **backuphash** file.

```
md5sum inventory.tar >> backuphash
```

NOTE: For this step, you must use `>>` to **append** the hash to the existing **backuphash** file. If you only use `>`, it will overwrite the existing file contents.

4. Run the **cat backuphash** command to display the two hashes. They should be the same.



```
(kali㉿kali)-[~]  
$ md5sum inventory.tar >> backuphash  
  
(kali㉿kali)-[~]  
$ cat backuphash  
534314a31037ac3bfee4bfb6b46dfd75  inventory.tar  
534314a31037ac3bfee4bfb6b46dfd75  inventory.tar
```

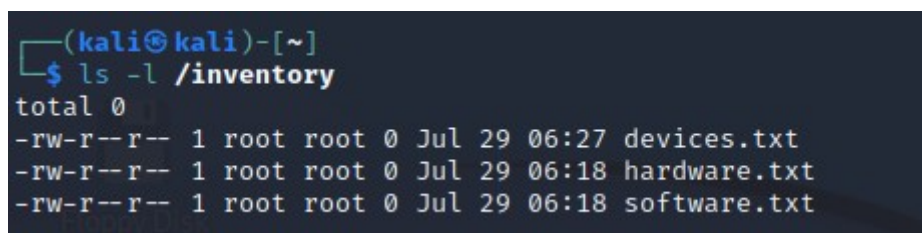
Figure 4.2 – Comparing the two md5sum hash values.

Because the two hash values are the same, you can be confident that the backup file did not change. The content you are restoring is identical to the content you backed up.

5. Run the following commands to extract the **inventory.tar** tarball to a directory named **/inventory**:

```
sudo tar -xf inventory.tar -C /
```

6. Display the **/inventory** directory to confirm that the restore process was successful.



```
(kali㉿kali)-[~]  
$ ls -l /inventory  
total 0  
-rw-r--r-- 1 root root 0 Jul 29 06:27 devices.txt  
-rw-r--r-- 1 root root 0 Jul 29 06:18 hardware.txt  
-rw-r--r-- 1 root root 0 Jul 29 06:18 software.txt
```

Figure 4.3 – Displaying the contents of /inventory folder.

You should see the **devices.txt**, **hardware.txt**, and **software.txt** files.