<u>Forensic Analysis Notes:</u>

Case Information:

Case: Extract the registry hives and scheduled tasks.

Analyst: Julius Niyonzima

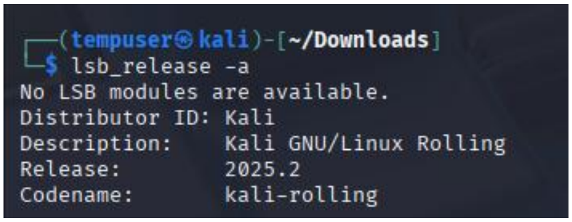
Professor: Dr. Chris Lamb

Email: jnbfg@umkc.edu

Id	File name	Hash(None)	Notes
1	Disk image	Zipped windows image	Suspicious Windows files need to be investigated

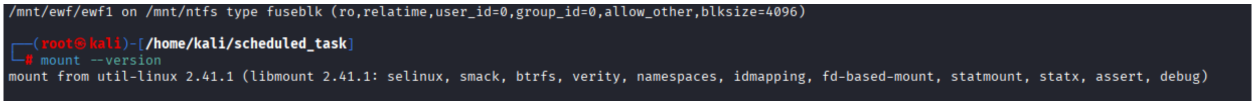
Tools and Environment

Kali Linux v2025.2



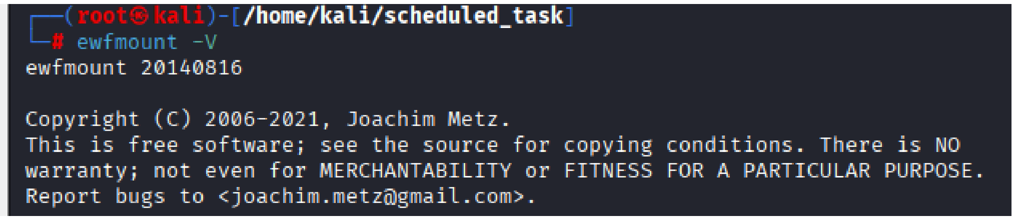
```
┌──(tempuser㉿kali)-[~/Downloads]
└─$ lsb_release -a
No LSB modules are available.
Distributor ID: Kali
Description:    Kali GNU/Linux Rolling
Release:        2025.2
Codename:       kali-rolling
```

mount v2.41.1 to attach filesystem like(USB, partition, hard drive, and ISO image) to existing directory structure



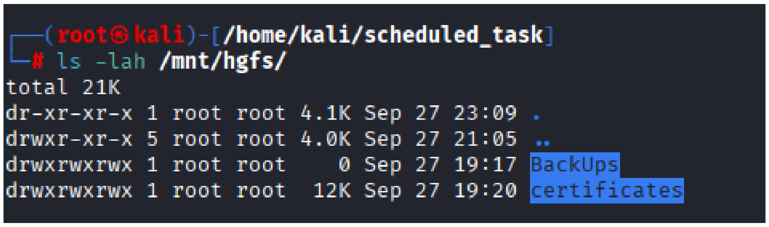
```
/mnt/ewf/ewf1 on /mnt/ntfs type fuseblk (ro,relatime,user_id=0,group_id=0,allow_other,blksize=4096)

┌──(root㉿kali)-[/home/kali/scheduled_task]
└─# mount --version
mount from util-linux 2.41.1 (libmount 2.41.1: selinux, smack, btrfs, verity, namespaces, idmapping, fd-based-mount, statmount, statx, assert, debug)
```

Ewfmount : Mount expert witness format (ewf) as a raw disk for analysis

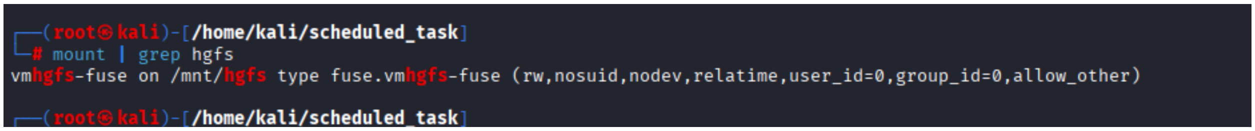


```
┌──(root㉿kali)-[/home/kali/scheduled_task]
└─# ewfmount -V
ewfmount 20140816

Copyright (C) 2006-2021, Joachim Metz.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
Report bugs to <joachim.metz@gmail.com>.
```

/mnt/hgfs/: created a shard folder to allow the virtual machine to access the folder on my external SanDisk driver

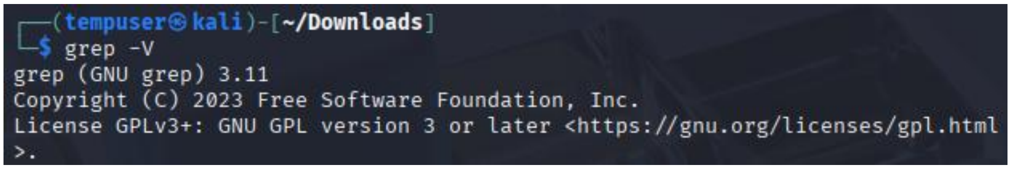


```
┌──(root㉿kali)-[/home/kali/scheduled_task]
└─# ls -lah /mnt/hgfs/
total 21K
dr-xr-xr-x 1 root root 4.1K Sep 27 23:09 .
drwxr-xr-x 5 root root 4.0K Sep 27 21:05 ..
drwxrwxrwx 1 root root    0 Sep 27 19:17 BackUps
drwxrwxrwx 1 root root  12K Sep 27 19:20 certificates
```



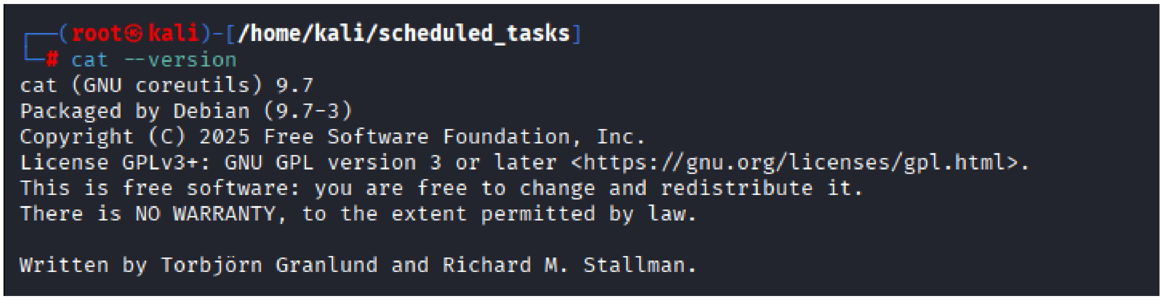
```
┌──(root㉿kali)-[/home/kali/scheduled_task]
└─# mount | grep hgfs
vmhgfs-fuse on /mnt/hgfs type fuse.vmhgfs-fuse (rw,nosuid,nodev,relatime,user_id=0,group_id=0,allow_other)

┌──(root㉿kali)-[/home/kali/scheduled_task]
```

Grep v3.11



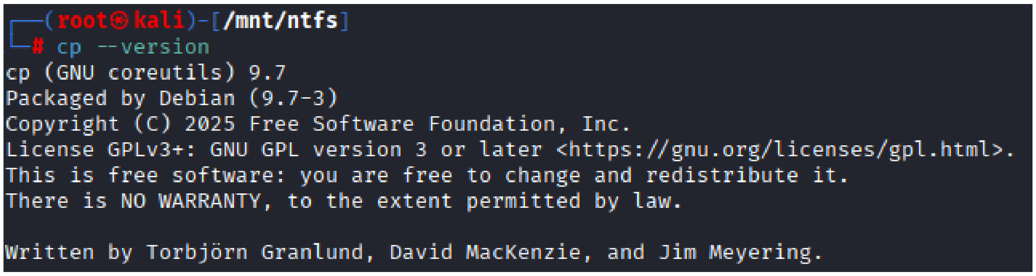
```
┌──(tempuser㉿kali)-[~/Downloads]
└─$ grep -V
grep (GNU grep) 3.11
Copyright (C) 2023 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <https://gnu.org/licenses/gpl.html
>.
```

Cat:



```
┌──(root㉿kali)-[/home/kali/scheduled_tasks]
└─# cat --version
cat (GNU coreutils) 9.7
Packaged by Debian (9.7-3)
Copyright (C) 2025 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <https://gnu.org/licenses/gpl.html>.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Written by Torbjörn Granlund and Richard M. Stallman.
```

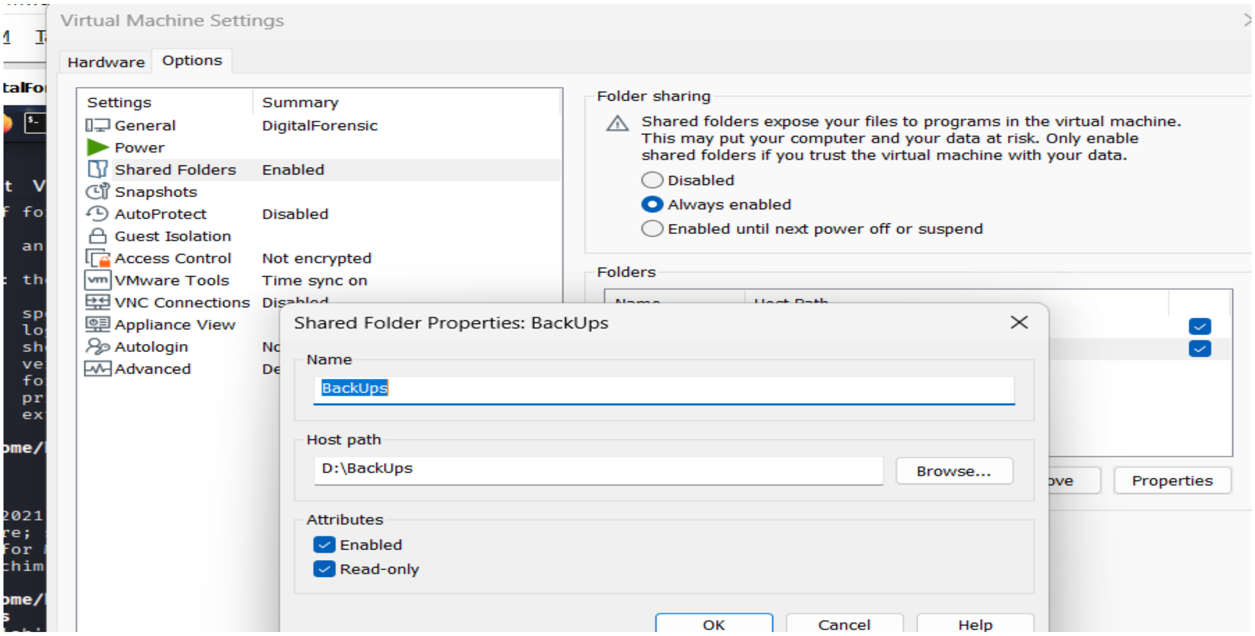
cop v9.7 for copying files and directories



```
┌──(root㉿kali)-[/mnt/ntfs]
└─# cp --version
cp (GNU coreutils) 9.7
Packaged by Debian (9.7-3)
Copyright (C) 2025 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <https://gnu.org/licenses/gpl.html>.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Written by Torbjörn Granlund, David MacKenzie, and Jim Meyering.
```

Reglookup command was used read and analyze data in windows registry hives

Step 1: Created a shared folder to allow the virtual machine (VM) to access the **backup** folder located at **D:\holders* on the external drive. This folder contains the disk image.



Settings	Summary
General	DigitalForensic
Power	
Shared Folders	Enabled
Snapshots	
AutoProtect	Disabled
Guest Isolation	
Access Control	Not encrypted
VMware Tools	Time sync on
VNC Connections	Disabled
Appliance View	
Autologin	No
Advanced	De

Folder sharing

Shared folders expose your files to programs in the virtual machine. This may put your computer and your data at risk. Only enable shared folders if you trust the virtual machine with your data.

○ Disabled
● Always enabled
○ Enabled until next power off or suspend

Folders

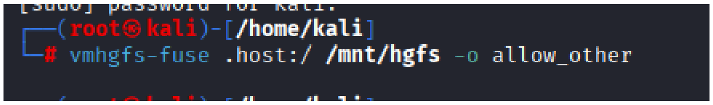
Shared Folder Properties: BackUps

Name: BackUps

Host path: D:\BackUps

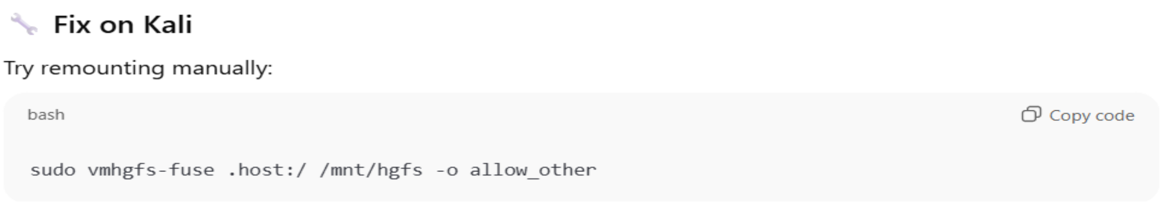
Attributes
☑ Enabled
☑ Read-only

Step 2: Mount manually to allow vm to share the shared folder



```
[sudo] password for kali:
┌──(root㉿kali)-[/home/kali]
└─# vmhgfs-fuse .host:/ /mnt/hgfs -o allow_other
```

copied from chatGPT



Fix on Kali

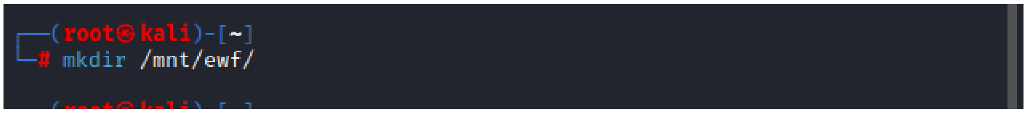
Try remounting manually:

```bash
sudo vmhgfs-fuse .host:/ /mnt/hgfs -o allow_other
```

Analysis Steps :

Step3 : Created the directory where the image should be mounted and used the **ewfmount** command to mount the image

Command:



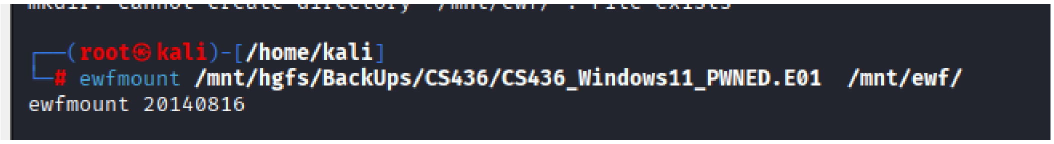
```
┌──(root㉿kali)-[~]
└─# mkdir /mnt/ewf/
```

Result: Directory was created

Status: Worked

Step4 : Mount the image to directory created

Command

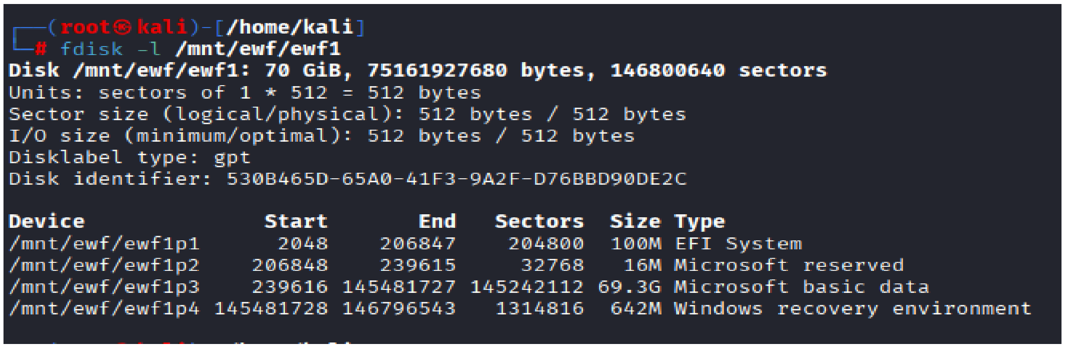


```
mkdir: cannot create directory '/mnt/ewf/': File exists

┌──(root㉿kali)-[/home/kali]
└─# ewfmount /mnt/hgfs/BackUps/CS436/CS436_Windows11_PWNED.E01  /mnt/ewf/
ewfmount 20140816
```

Result: The entire image was mounted in the /mnt/ewf/ directory

Status: worked

Step 5: List the characteristics of the mounted drive



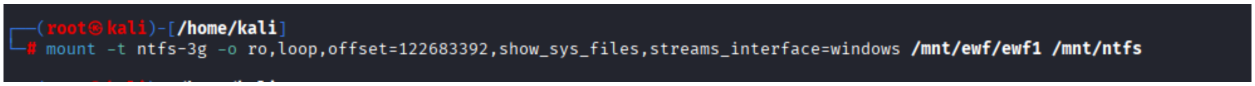
```
┌──(root㉿kali)-[/home/kali]
└─# fdisk -l /mnt/ewf/ewf1
Disk /mnt/ewf/ewf1: 70 GiB, 75161927680 bytes, 146800640 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: 530B465D-65A0-41F3-9A2F-D76BBD90DE2C

Device              Start       End   Sectors  Size Type
/mnt/ewf/ewf1p1      2048    206847    204800  100M EFI System
/mnt/ewf/ewf1p2    206848    239615     32768   16M Microsoft reserved
/mnt/ewf/ewf1p3    239616 145481727 145242112 69.3G Microsoft basic data
/mnt/ewf/ewf1p4 145481728 146796543   1314816  642M Windows recovery environment
```

Result: Partitions captured were listed, with the sector starting **at 239616**

Status: Worked

Step 5: Mount the NTFS partitions

Command:

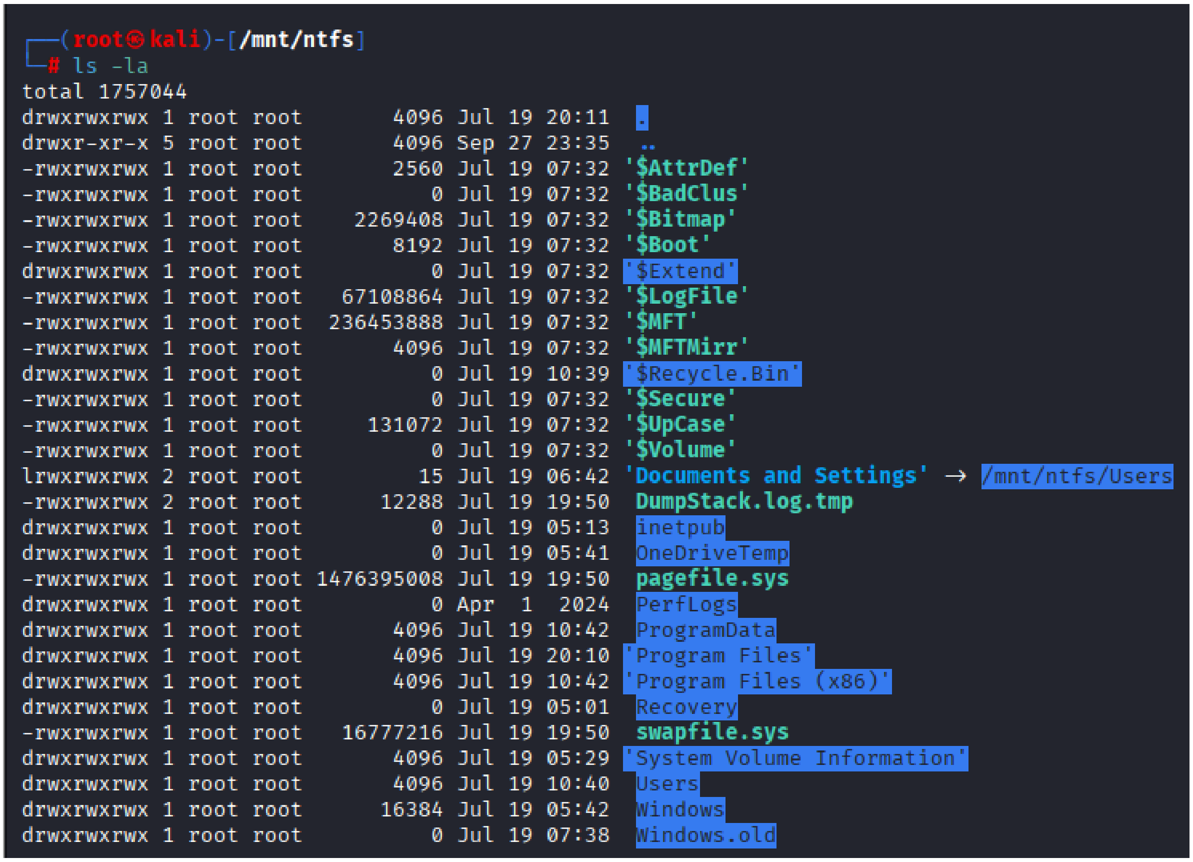


```
┌──(root㉿kali)-[/home/kali]
└─# mount -t ntfs-3g -o ro,loop,offset=122683392,show_sys_files,streams_interface=windows /mnt/ewf/ewf1 /mnt/ntfs
```

Result: The disk image was mounted as read-only at /mnt/ntfs

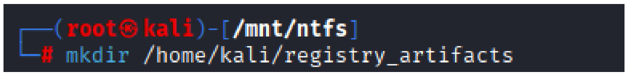
Status: Worked

Step 6: listed all contents from the disk image mounted at /mnt/ntfs



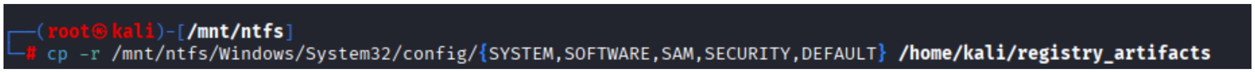
```
┌──(root㉿kali)-[/mnt/ntfs]
└─# ls -la
total 1757044
drwxrwxrwx 1 root root       4096 Jul 19 20:11  .
drwxr-xr-x 5 root root       4096 Sep 27 23:35  ..
-rwxrwxrwx 1 root root       2560 Jul 19 07:32 '$AttrDef'
-rwxrwxrwx 1 root root          0 Jul 19 07:32 '$BadClus'
-rwxrwxrwx 1 root root    2269408 Jul 19 07:32 '$Bitmap'
-rwxrwxrwx 1 root root       8192 Jul 19 07:32 '$Boot'
drwxrwxrwx 1 root root          0 Jul 19 07:32 '$Extend'
-rwxrwxrwx 1 root root   67108864 Jul 19 07:32 '$LogFile'
-rwxrwxrwx 1 root root  236453888 Jul 19 07:32 '$MFT'
-rwxrwxrwx 1 root root       4096 Jul 19 07:32 '$MFTMirr'
drwxrwxrwx 1 root root          0 Jul 19 10:39 '$Recycle.Bin'
-rwxrwxrwx 1 root root          0 Jul 19 07:32 '$Secure'
-rwxrwxrwx 1 root root     131072 Jul 19 07:32 '$UpCase'
-rwxrwxrwx 1 root root          0 Jul 19 07:32 '$Volume'
lrwxrwxrwx 2 root root         15 Jul 19 06:42 'Documents and Settings' → /mnt/ntfs/Users
-rwxrwxrwx 2 root root      12288 Jul 19 19:50  DumpStack.log.tmp
drwxrwxrwx 1 root root          0 Jul 19 05:13  inetpub
drwxrwxrwx 1 root root          0 Jul 19 05:41  OneDriveTemp
-rwxrwxrwx 1 root root 1476395008 Jul 19 19:50  pagefile.sys
drwxrwxrwx 1 root root          0 Apr  1  2024  PerfLogs
drwxrwxrwx 1 root root       4096 Jul 19 10:42  ProgramData
drwxrwxrwx 1 root root       4096 Jul 19 20:10 'Program Files'
drwxrwxrwx 1 root root       4096 Jul 19 10:42 'Program Files (x86)'
drwxrwxrwx 1 root root          0 Jul 19 05:01  Recovery
-rwxrwxrwx 1 root root   16777216 Jul 19 19:50  swapfile.sys
drwxrwxrwx 1 root root       4096 Jul 19 05:29 'System Volume Information'
drwxrwxrwx 1 root root       4096 Jul 19 10:40  Users
drwxrwxrwx 1 root root      16384 Jul 19 05:42  Windows
drwxrwxrwx 1 root root          0 Jul 19 07:38  Windows.old
```

Step 7: I made a directory to hold all registry artifacts



```
┌──(root㉿kali)-[/mnt/ntfs]
└─# mkdir /home/kali/registry_artifacts
```

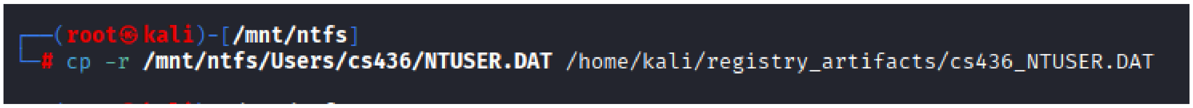
Step 8: All the hives were copied from the image to the new directory, one by one



```
┌──(root㉿kali)-[/mnt/ntfs]
└─# cp -r /mnt/ntfs/Windows/System32/config/{SYSTEM,SOFTWARE,SAM,SECURITY,DEFAULT} /home/kali/registry_artifacts
```

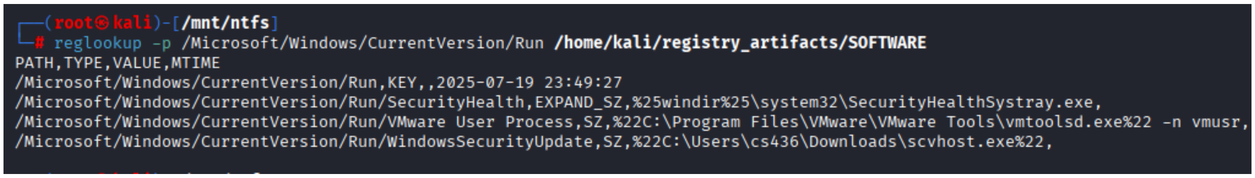
Step 8: The NTUSER.DAT file that stores user-specific Windows registry settings was copied to the Cs436_NTUSER directory

Command:



```
┌──(root㉿kali)-[/mnt/ntfs]
└─# cp -r /mnt/ntfs/Users/cs436/NTUSER.DAT /home/kali/registry_artifacts/cs436_NTUSER.DAT
```

Step 9: Read and analyze data about applications configured to run at Windows startup



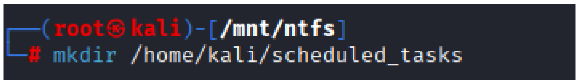
```
┌──(root㉿kali)-[/mnt/ntfs]
└─# reglookup -p /Microsoft/Windows/CurrentVersion/Run /home/kali/registry_artifacts/SOFTWARE
PATH,TYPE,VALUE,MTIME
/Microsoft/Windows/CurrentVersion/Run,KEY,,2025-07-19 23:49:27
/Microsoft/Windows/CurrentVersion/Run/SecurityHealth,EXPAND_SZ,%25windir%25\system32\SecurityHealthSystray.exe,
/Microsoft/Windows/CurrentVersion/Run/VMware User Process,SZ,%22C:\Program Files\VMware\VMware Tools\vmtoolsd.exe%22 -n vmusr,
/Microsoft/Windows/CurrentVersion/Run/WindowsSecurityUpdate,SZ,%22C:\Users\cs436\Downloads\scvhost.exe%22,
```

Result: Under WindowsSecurityUpdate, there is a suspicious executable designed to run at Windows startup called scvhost.exe, instead of the usual Windows svchost.exe. This could be malware or another form of suspicious persistence activity.

Status: worked

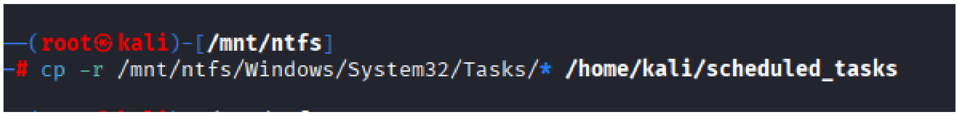
Step 10: Make a directory to handle scheduled tasks.

Command:



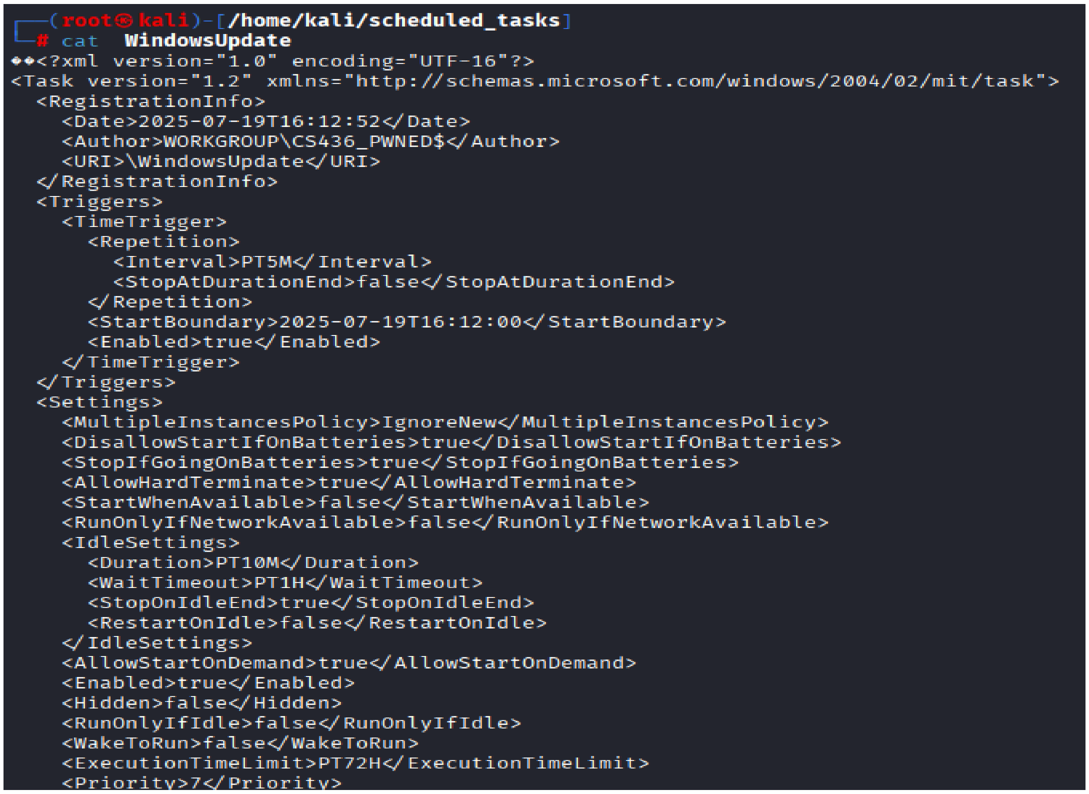
```
┌──(root㉿kali)-[/mnt/ntfs]
└─# mkdir /home/kali/scheduled_tasks
```

Step 11: Copying all scheduled tasks stored in Windows/System32/Tasks

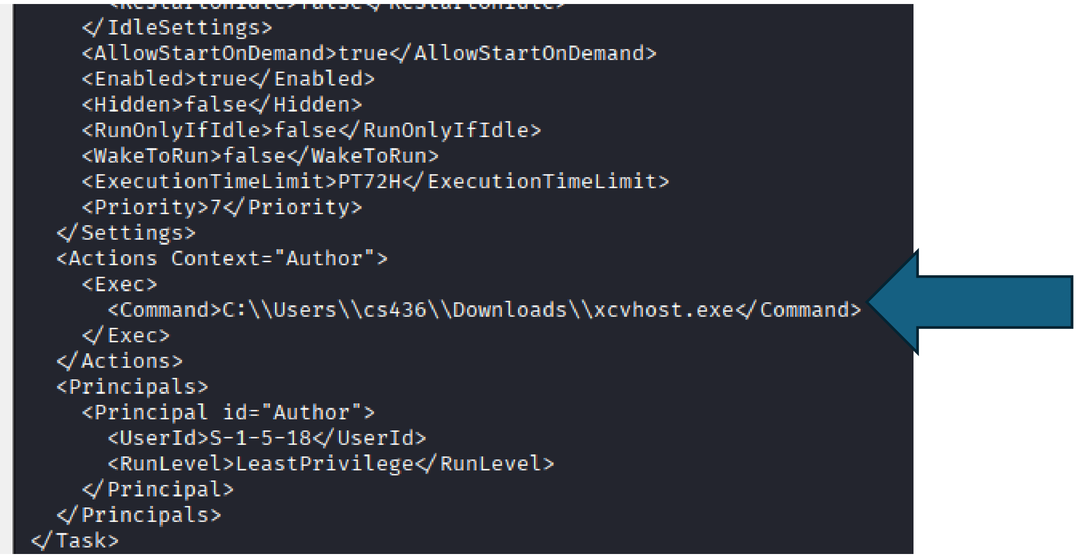


```
──(root㉿kali)-[/mnt/ntfs]
─# cp -r /mnt/ntfs/Windows/System32/Tasks/* /home/kali/scheduled_tasks
```

Step 12: Find an XML file that contains a command to be executed at Windows startup



```
┌──(root㉿kali)-[/home/kali/scheduled_tasks]
└─# cat  WindowsUpdate
��<?xml version="1.0" encoding="UTF-16"?>
<Task version="1.2" xmlns="http://schemas.microsoft.com/windows/2004/02/mit/task">
  <RegistrationInfo>
    <Date>2025-07-19T16:12:52</Date>
    <Author>WORKGROUP\CS436_PWNED$</Author>
    <URI>\WindowsUpdate</URI>
  </RegistrationInfo>
  <Triggers>
    <TimeTrigger>
      <Repetition>
        <Interval>PT5M</Interval>
        <StopAtDurationEnd>false</StopAtDurationEnd>
      </Repetition>
      <StartBoundary>2025-07-19T16:12:00</StartBoundary>
      <Enabled>true</Enabled>
    </TimeTrigger>
  </Triggers>
  <Settings>
    <MultipleInstancesPolicy>IgnoreNew</MultipleInstancesPolicy>
    <DisallowStartIfOnBatteries>true</DisallowStartIfOnBatteries>
    <StopIfGoingOnBatteries>true</StopIfGoingOnBatteries>
    <AllowHardTerminate>true</AllowHardTerminate>
    <StartWhenAvailable>false</StartWhenAvailable>
    <RunOnlyIfNetworkAvailable>false</RunOnlyIfNetworkAvailable>
    <IdleSettings>
      <Duration>PT10M</Duration>
      <WaitTimeout>PT1H</WaitTimeout>
      <StopOnIdleEnd>true</StopOnIdleEnd>
      <RestartOnIdle>false</RestartOnIdle>
    </IdleSettings>
    <AllowStartOnDemand>true</AllowStartOnDemand>
    <Enabled>true</Enabled>
    <Hidden>false</Hidden>
    <RunOnlyIfIdle>false</RunOnlyIfIdle>
    <WakeToRun>false</WakeToRun>
    <ExecutionTimeLimit>PT72H</ExecutionTimeLimit>
    <Priority>7</Priority>
```



```
      <RestartOnIdle>false</RestartOnIdle>
    </IdleSettings>
    <AllowStartOnDemand>true</AllowStartOnDemand>
    <Enabled>true</Enabled>
    <Hidden>false</Hidden>
    <RunOnlyIfIdle>false</RunOnlyIfIdle>
    <WakeToRun>false</WakeToRun>
    <ExecutionTimeLimit>PT72H</ExecutionTimeLimit>
    <Priority>7</Priority>
  </Settings>
  <Actions Context="Author">
    <Exec>
      <Command>C:\\Users\\cs436\\Downloads\\xcvhost.exe</Command>
    </Exec>
  </Actions>
  <Principals>
    <Principal id="Author">
      <UserId>S-1-5-18</UserId>
      <RunLevel>LeastPrivilege</RunLevel>
    </Principal>
  </Principals>
</Task>
```

Result: Commands embedded in the XML files are set to run at Windows startup, which may indicate malware or suspicious persistence activity.

Findings:

The 'WindowsSecurityUpdate' key contained a misspelled executable, *scvhost.exe*, set to run at Windows startup. Additionally, an XML file named *windowsUpdate* was found, embedded with a command pointing to C:\Users\cs436\Downloads\xcvhost.exe, which is also configured to run at startup. This behavior appears suspicious and may indicate malware or a persistence mechanism.

Conclusion

The presence of a misspelled executable (scvhost.exe) configured to run at Windows startup via the WindowsSecurityUpdate registry key, along with an XML file (windowsUpdate) that contains a command to execute scvhost.exe from the user’s Downloads folder. May suggests suspicious activity. With all these artifacts, they exhibit common characteristics of **malware or persistence mechanisms** often used to maintain unauthorized access to a system. A more detailed investigation and analysis of malware is recommended to confirm the existence and nature of these files.