Advanced Bash: Owning the System

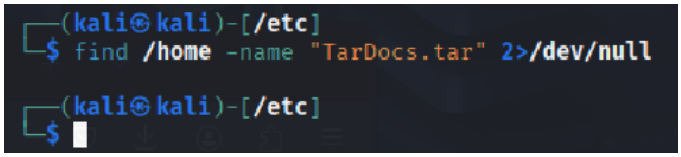
Category: System Hardening & Endpoint Security
Tagline: Securing Linux environments by automating system monitoring, managing logs, and configuring auditing tools for integrity.

1. Archiving Files While Excluding Certain Directories

1.1: Locate the Archive

bash

```
find /home -name "TarDocs.tar" 2>/dev/null
```

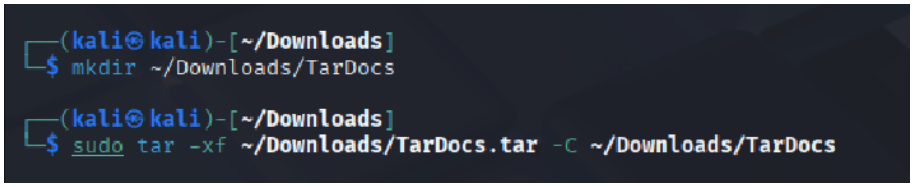


Why: Searches for the archive file across /home, hiding any "Permission Denied" errors.

1.2: Extract the Archive into a Dedicated Folder

bash

```
mkdir ~/Downloads/TarDocs
sudo tar -xf ~/Downloads/TarDocs.tar -C ~/Downloads/TarDocs
```

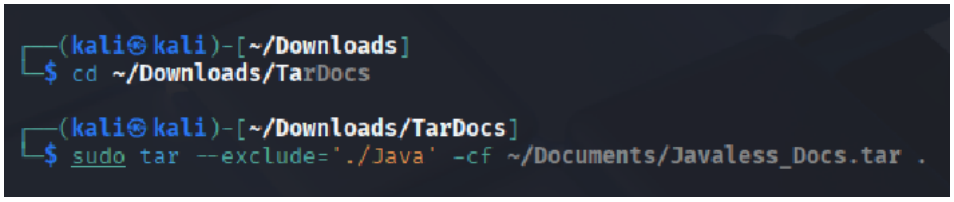


Unpacks the contents into your `Documents` directory for review or modification.

1.3: Create a New Archive Excluding the `Java` Folder

bash

```
cd ~/Downloads/TarDocs
sudo tar --exclude='./Java' -cf ~/Documents/Javaless_Docs.tar .
```



Why: Creates a new archive without the `Java` folder and stores it in the `Documents` directory to avoid self-inclusion errors.

1.4: Verify the Archive Was Created Successfully

bash

```
ls -lh ~/Documents/Javaless_Docs.tar
```

```
┌──(kali㉿kali)-[~/Downloads/TarDocs]
└─$ ls -lh ~/Documents/Javaless_Docs.tar
-rw-r--r-- 1 root root 1.1G Jul 22 16:08 /home/kali/Documents/Javaless_Docs.tar
```

Why: Confirms the new archive exists and shows its file size.

1.5: Confirm the Java Folder Is Not in the Archive

bash

```
tar -tf ~/Documents/Javaless_Docs.tar | grep Java
```

```
┌──(kali㉿kali)-[~/Downloads/TarDocs]
└─$ tar -tf ~/Documents/Javaless_Docs.tar | grep Java
./TarDocs/Documents/Java/
./TarDocs/Documents/Java/Java-Network-Programming-3e/
./TarDocs/Documents/Java/Java-Network-Programming-3e/chp-18-RMI.pdf
./TarDocs/Documents/Java/Java-Network-Programming-3e/chp-17-Content Handle
```

Why: Lists the contents of the archive and confirms that the `Java` directory was excluded. If no output appears, exclusion was successful.

2. Create and Schedule System Reports with Cron

2.1: Open Crontab

bash

```
crontab -e
```

```
┌──(kali㉿kali)-[~/Downloads/TarDocs]
└─$ crontab -e
no crontab for kali - using an empty one
Select an editor.  To change later, run select-editor again.
  1. /bin/nano        <---- easiest
  2. /usr/bin/vim.basic
  3. /usr/bin/vim.tiny

Choose 1-3 [1]: 1
```

Opens the crontab editor where you can define scheduled tasks.

2.2: Schedule a Daily System Report

bash

```
@daily ~/scripts/system.sh > ~/backups/system_report_$(date +\%F).txt
```

Runs the `system.sh` script daily and saves the output with the current date.

```
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
@daily ~/scripts/system.sh > ~/backups/system_report_$(date +\%F).txt
```

3. Organize Script Outputs by Type

3.1: Create Backup Subdirectories

bash

```
mkdir -p ~/backups/{freemem,diskuse,openlist,freedisk}
```

```
┌──(kali㉿kali)-[~/Downloads/TarDocs]
└─$ mkdir -p ~/backups/{freemem,diskuse,openlist,freedisk}
```

Organizes backups by creating subdirectories for each type of system report.

4. Implement File Auditing Using auditd

4.1: Install auditd

bash

```
sudo apt update && sudo apt install auditd -y
```

```
┌──(kali㉿kali)-[~/Downloads/TarDocs]
└─$ sudo apt update && sudo apt install auditd -y
Hit:1 http://http.kali.org/kali kali-rolling InRelease
1297 packages can be upgraded. Run 'apt list --upgradable' to see them.
auditd is already the newest version (1:4.0.2-2+b2).
Summary:
  Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 1297
```

Installs the auditing daemon for tracking file and system changes.

4.2: Start and Enable auditd

bash

```
sudo systemctl enable --now auditd
```

```
┌──(kali㉿kali)-[~/Downloads/TarDocs]
└─$ sudo systemctl enable --now auditd
Synchronizing state of auditd.service with SysV service script with /usr/l
ib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable auditd
```

Enables and starts auditd to persist across reboots.

4.3: Verify auditd Status

bash

```
sudo systemctl status auditd
```

```
┌──(kali㉿kali)-[~/Downloads/TarDocs]
└─$ sudo systemctl status auditd
● auditd.service - Security Audit Logging Service
     Loaded: loaded (/usr/lib/systemd/system/auditd.service; enabled; pre>
     Active: active (running) since Tue 2025-07-22 11:42:31 PDT; 4h 42min>
 Invocation: ebf67489ccc241d0b822c00a4c3e66bd
       Docs: man:auditd(8)
             https://github.com/linux-audit/audit-documentation
   Main PID: 733136 (auditd)
      Tasks: 2 (limit: 4490)
     Memory: 1.4M (peak: 1.9M)
        CPU: 140ms
     CGroup: /system.slice/auditd.service
             └─733136 /usr/sbin/auditd

Jul 22 11:42:31 kali systemd[1]: Starting auditd.service - Security Audit>
Jul 22 11:42:31 kali auditd[733136]: No plugins found, not dispatching ev>
Jul 22 11:42:31 kali auditd[733136]: Init complete, auditd 4.0.2 listenin>
Jul 22 11:42:31 kali systemd[1]: Started auditd.service - Security Audit >
lines 1-17/17 (END)
```

Confirms auditd is running properly.

4.4: Monitor cron File for Changes

bash

```
sudo auditctl -w /var/log/cron -p rwa -k cron_watch
```

```
┌──(kali㉿kali)-[~/Downloads/TarDocs]
└─$ sudo auditctl -w /var/log/cron -p rwa -k cron_watch
Old style watch rules are slower
```

Watches the cron log file for read, write, and attribute changes with key `cron_watch`.

4.5: List All auditd Rules

bash

```
sudo auditctl -l
```

```
┌──(kali㉿kali)-[~/Downloads/TarDocs]
└─$ sudo auditctl -l
-w /etc/passwd -p rwa -k hashpass_audit
-w /etc/passwd -p rwa -k userpass_audit
-w /var/log/auth.log -p rwa -k authlog_audit
-w /var/log/cron -p rwa -k cron_watch
```

Displays all active audit rules to verify configuration.

4.6: Search Logs by Key

bash

```
sudo ausearch -k cron_watch
```

```
┌──(kali㉿kali)-[~/Downloads/TarDocs]
└─$ sudo ausearch -k cron_watch
----
time->Tue Jul 22 11:51:47 2025
type=PROCTITLE msg=audit(1753210307.841:101): proctitle=617564697463746C00
2D77002F7661722F6C6F672F63726F6E002D7000777261002D6B0063726F6E5F7761746368
type=PATH msg=audit(1753210307.841:101): item=0 name="/var/log/" inode=524
448 dev=fe:03 mode=040755 ouid=0 ogid=0 rdev=00:00 nametype=PARENT cap_fp=
0 cap_fi=0 cap_fe=0 cap_fver=0 cap_frootid=0
type=CWD msg=audit(1753210307.841:101): cwd="/root"
type=SOCKADDR msg=audit(1753210307.841:101): saddr=100000000000000000000000
0
```

Queries logs that triggered the `cron_watch` rule.

5. Create a System Resource Usage Script

5.1: Create system.sh

bash

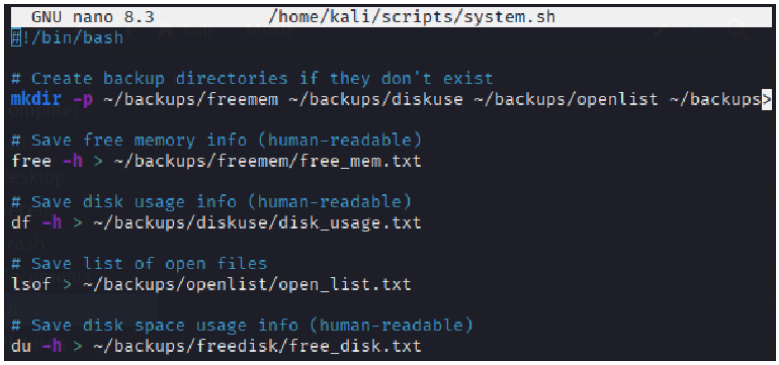
```
nano ~/scripts/system.sh
```

5.2: Write the Bash Script Content

Inside the nano editor, enter the following script to gather system resource usage and save outputs to organized backup directories:

bash

```bash
#!/bin/bash

# Create backup directories if they don't exist
mkdir -p ~/backups/freemem ~/backups/diskuse ~/backups/openlist
~/backups/freedisk

# Save free memory info (human-readable)
free -h > ~/backups/freemem/free_mem.txt

# Save disk usage info (human-readable)
df -h > ~/backups/diskuse/disk_usage.txt

# Save list of open files
lsof > ~/backups/openlist/open_list.txt

# Save disk space usage info (human-readable)
du -h > ~/backups/freedisk/free_disk.txt
```



5.3: Save and Exit

- Press `Ctrl + O` to save the file.
- Press `Enter` to confirm.
- Press `Ctrl + X` to exit nano.

5.4: Make the Script Executable

bash

```bash
chmod +x ~/scripts/system.sh
```

5.5: Run the Script to Test

bash

```
~/scripts/system.sh
```

```
┌──(kali㉿kali)-[~/Downloads/TarDocs]
└─$ ~/scripts/system.sh
```

5.6: Verify Output Files

Check that the output files are created and contain data, for example:

bash

```
cat ~/backups/freemem/free_mem.txt
```

```
┌──(kali㉿kali)-[~/Downloads/TarDocs]
└─$ cat ~/backups/freemem/free_mem.txt
              total        used        free      shared  buff/cache   ava
ilable
Mem:          3.8Gi       2.1Gi       201Mi        58Mi       1.9Gi
 1.8Gi
Swap:         1.6Gi       163Mi       1.4Gi
```

```
cat ~/backups/diskuse/disk_usage.txt
```

```
┌──(kali㉿kali)-[~/Downloads/TarDocs]
└─$ cat ~/backups/diskuse/disk_usage.txt
Filesystem      Size  Used Avail Use% Mounted on
udev            1.9G     0  1.9G   0% /dev
tmpfs           392M  1.3M  390M   1% /run
/dev/vda3        27G   22G  4.4G  84% /
tmpfs           2.0G  4.0K  2.0G   1% /dev/shm
efivarfs        256K   26K  231K  11% /sys/firmware/efi/efivars
tmpfs           5.0M     0  5.0M   0% /run/lock
tmpfs           1.0M     0  1.0M   0% /run/credentials/systemd-journald.se
rvice
tmpfs           2.0G  680K  2.0G   1% /tmp
/dev/vda2       977M  192K  977M   1% /boot/efi
tmpfs           1.0M     0  1.0M   0% /run/credentials/getty@tty1.service
tmpfs           1.0M     0  1.0M   0% /run/credentials/serial-getty@ttyAMA
0.service
tmpfs           392M  120K  391M   1% /run/user/1000
```