dmf lab

1. Installation of Sleuth Kit on Linux

Aim:

To install Sleuth Kit on Linux and analyze allocated & unallocated blocks of a disk image.

Procedure:

1. Open Terminal → Install Sleuth Kit using `sudo apt install sleuthkit`.
2. Create a disk image using `sudo dd if=/dev/sdb of=diskimage.dd bs=4M`.
3. View image info: `img_stat diskimage.dd`.
4. List data blocks: `mmls diskimage.dd`.
5. Use `fsstat` and `fls` to analyze allocated partitions.
6. Analyze unallocated partitions using `icat` and recover deleted files.

Diagram:

A simple flow:

```
Disk → Image Creation → mmls → fsstat/fls → icat (recovery)
```

Result:

Sleuth Kit successfully installed and disk image blocks analyzed.

2. Installation of Sleuth Kit and List All Data Blocks

Aim:

To list all data blocks using Sleuth Kit tools.

Procedure:

1. Create disk image using `dd`.

2. Use `mmls` to list block details (start, end, size).
3. Identify unallocated spaces.
4. Verify partitions using `fsstat`.

Diagram:

```
Disk Image → mmls → Block Table Output
```

Result:

All allocated and unallocated blocks listed successfully.

---

3. Analyze Allocated and Unallocated Blocks

Aim:

To analyze both allocated and unallocated blocks using Sleuth Kit.

Procedure:

1. Use `mmls` to identify partitions.
2. Apply `fls` on allocated area to list files.
3. Use `icat` to read deleted/unallocated files.
4. Compare results using recovered file metadata.

Diagram:

```
Allocated → fls
Unallocated → icat → File Recovery
```

Result:

Allocated and unallocated areas analyzed successfully.

---

4. Allocate a Disk Image Using Sleuth Kit

Aim:

To allocate a disk image for forensic analysis.

Procedure:

1. Install Sleuth Kit.
2. Create a raw disk image (`.dd`).
3. Use `mmls` to assign partition boundaries.
4. Mount and analyze image with Autopsy or Sleuth Kit.

Diagram:

```
Disk → Image (.dd) → Allocation → Analysis
```

Result:

Disk image successfully allocated and ready for analysis.

---

5. Data Extraction from Call Logs Using Sleuth Kit

Aim:

To extract call log information using Sleuth Kit (Autopsy interface).

Procedure:

1. Open Autopsy → Create New Case.
2. Add the phone image file.
3. Select modules (Applications, Contacts, Call Logs).
4. View extracted call records → Export CSV.

Diagram:

```
Autopsy → Case → Modules → Call Logs → Export CSV
```

Result:

Call logs extracted and viewed successfully.

---

6. Allocate Disk Image and Extract Call Logs

Aim:

To allocate a phone image and extract call logs.

Procedure:

1. Use Sleuth Kit tools to mount disk image.
2. Import image into Autopsy.
3. Run extraction modules for call logs.
4. Export and view call data report.

Diagram:

```
Disk Image → Autopsy → Call Logs Extraction
```

Result:

Call log extraction from allocated disk image completed.

---

7. Data Extraction from SMS and Contacts Using Sleuth Kit

Aim:

To extract SMS and contacts from phone image.

Procedure:

1. Open Autopsy → Load image.
2. Select `sms.db` and `contacts.db`.
3. Open in SQLite browser → Export to CSV.
4. Analyze SMS and contact records.

Diagram:

```
Autopsy → SQLite → SMS / Contacts → CSV
```

Result:

SMS and contact data extracted successfully.

---

8. Install Sleuth Kit and Create SMS & Contacts

Aim:

To create and analyze sample SMS and contact data.

Procedure:

1. Install Sleuth Kit.
2. Create sample `sms.db` and `contacts.db`.
3. Import into Autopsy for analysis.
4. Generate CSV report.

Diagram:

```
Create DB → Autopsy → Analyze → Report
```

Result:

Sample SMS and contact data created and analyzed.

---

9. Create Data and Allocate Disk Image

Aim:

To create sample data and allocate disk image for forensic testing.

Procedure:

1. Create folder with sample files.
2. Generate image using `dd`.
3. Allocate partitions using `mmls`.
4. Verify structure using `fsstat`.

Diagram:

```
Files → dd Image → mmls → fsstat
```

Result:

Data successfully created and allocated for Sleuth Kit.

---

10. Install Autopsy and Create Dataset

Aim:

To install Autopsy forensic tool and create dataset for analysis.

Procedure:

1. Install using `sudo apt install autopsy`.
2. Launch via `autopsy &`.
3. Create new case → add data source.
4. Run modules for dataset creation.

Diagram:

```
Autopsy → Case Creation → Data Source → Dataset
```

Result:

Autopsy installed and dataset created successfully.

---

11. Data Extraction in Autopsy Tool

Aim:

To extract digital evidence using Autopsy.

Procedure:

1. Load image file in Autopsy.
2. Choose “Keyword Search” or “File Analysis”.
3. Extract and export relevant artifacts.

4. Save results in report format.

Diagram:

```
Autopsy → Analysis → Extract → Report
```

Result:

Data extracted successfully using Autopsy.

---

12. Install Mobile Verification Toolkit (MVT)

Aim:

To install MVT for analyzing iOS/Android backups.

Procedure:

1. Install dependencies: `python3`, `pip`.
2. Run `pip install mvt`.
3. Test using `mvt-ios --help` or `mvt-android --help`.

Diagram:

```
Dependencies → pip install → Verification
```

Result:

MVT installed successfully on the system.

---

13. Reinstall or Update MVT

Aim:

To update or reinstall Mobile Verification Toolkit.

Procedure:

1. Run `pip uninstall mvt`.
2. Install latest version using `pip install mvt`.

3. Confirm version using `mvt --version`.

Diagram:

```
Uninstall → Install → Verify
```

Result:

MVT reinstalled and verified successfully.

---

14. Process and Parse Records from iOS System

Aim:

To parse and process iOS data records for analysis.

Procedure:

1. Identify iOS data types (contacts, SMS, logs).
2. Open `sqlite` and `plist` files using browser/editor.
3. Parse and export readable data formats.
4. Combine extracted records for analysis.

Diagram:

```
iOS Backup → SQLite/Plist → Parse → CSV/JSON
```

Result:

iOS records processed and parsed successfully.

---

15. Extract Installed Apps from Android

Aim:

To extract APKs of installed applications via ADB.

Procedure:

1. Enable Developer Options → USB Debugging.

2. Connect via ADB → `adb devices`.
3. List packages: `adb shell pm list packages -f`.
4. Extract APK using `adb pull`.

Diagram:

```
Phone → ADB → List Apps → Extract APK
```

Result:

All installed apps extracted successfully.

---

16. Extract Diagnostic Info Using ADB

Aim:

To extract system diagnostics from Android device.

Procedure:

1. Connect via ADB.
2. Run `adb bugreport` and `adb logcat`.
3. Save output as text files for analysis.

Diagram:

```
Android → ADB → Logs → Analysis
```

Result:

Diagnostic information extracted successfully.

---

17. Generate Unified Timeline of Extracted Records

Aim:

To merge multiple extracted logs into a single timeline.

Procedure:

1. Gather all log files (call, SMS, system).
2. Sort them by timestamp.
3. Use Excel or forensic tool to plot events chronologically.

Diagram:

```
Logs → Merge → Sort by Time → Timeline
```

Result:

Unified chronological timeline generated successfully.

---

18. Extract Installed Applications (Repeat)

Aim:

To verify and extract Android installed apps.

Procedure:

1. Connect phone via USB.
2. Use `adb shell pm list packages`.
3. Extract required APKs.

Diagram:

```
ADB → List → Extract
```

Result:

Applications extracted and verified successfully.

---

19. Extract Diagnostic Information (Repeat)

Aim:

To collect Android system diagnostics using ADB.

Procedure:

1. Use `adb shell dumpsys`.
2. Save results with `adb bugreport > diag.txt`.
3. Review reports for system health info.

Diagram:

```
ADB → Dumpsys → Bugreport → Analysis
```

Result:

Diagnostic report generated successfully.

---

20. Generate Unified Timeline

Aim:

To create a combined timeline of mobile activities.

Procedure:

1. Combine event logs (SMS, calls, app use).
2. Use spreadsheet or tool to arrange chronologically.
3. Analyze sequence of events.

Diagram:

```
Extracted Data → Merge → Time Sort → Timeline Chart
```

Result:

Unified chronological timeline successfully generated.
