Digital File Analysis and Information Extraction

Cyber Forensics Workshop

April 13, 2025

Contents

1 Introduction to Digital File Analysis

This section provides an overview of Digital File Analysis, which is essential in digital forensic investigations for uncovering evidence within suspicious files.
Key Points:

- Digital File Analysis focuses on identifying, extracting, and interpreting information from various file types.
- It is crucial in forensic investigations to ensure evidence is properly discovered and handled.
- Tools discussed include: `file`, `binwalk`, `strings`, `exiftool`, `xxd`, and more.

Primary Goals:

- Recognize file types, even if obfuscated.
- Extract metadata and hidden data from suspicious files.
- Understand common techniques for file protection and how to recover locked or encrypted files.

2 Understanding File Types and Magic Numbers

File signatures or Magic Numbers are byte patterns at the beginning of files, indicating the true file type regardless of the extension.

Common Tools

- `file`: Detects file type by reading magic bytes.
- `xxd`: Displays the hexadecimal dump of a file.

Example Commands:

```
file unknown_file.xyz
xxd unknown_file.xyz | head
```

Typical Magic Numbers:

- JPG: FF D8 FF
- PNG: 89 50 4E 47
- PDF:

Even if an extension is changed (e.g., `.jpg` renamed to `.txt`), analyzing magic numbers gives a reliable indication of the real file type.

3 Extracting Metadata from Images Using ExifTool

Images can contain important metadata that may reveal:

- Camera make and model
- Timestamps (date/time)
- GPS coordinates (if geotagging is enabled)

ExifTool

`exiftool` can read and write metadata for various file types, especially images:

```
exiftool image.jpg
```

Output may include fields like `Make`, `Model`, `DateTimeOriginal`, and `GPSLatitude/Longitude`.

4 Binary File Inspection with Strings and Binwalk

Sometimes, binaries or firmware images contain hidden or suspicious information.

strings

The `strings` command extracts readable text (ASCII or Unicode) from binary files:

```
strings suspicious_file.bin | less
```

This can reveal URLs, file paths, or error messages embedded in the file.

binwalk

`binwalk` analyzes a binary file to find embedded files, compressed data, or firmware components:

```
binwalk suspicious_file.bin
binwalk -e suspicious_file.bin  # auto-extract
```

5 Types of File Protection and Encryption Methods

Various file formats include security or encryption to protect content:

- Password-protected ZIP archives
- Encrypted PDF documents
- Word/Excel documents with passwords
- Full-disk or file-level encryption

Forensic analysis requires tools that can handle password recovery or hash extraction.

6 Password Cracking Techniques for Protected Files

Several methods to retrieve or crack passwords:

- **Brute Force**: Tries every possible password.
- **Dictionary Attack**: Uses a known or common-password wordlist (e.g., `rockyou.txt`).
- **Rule-based Attack**: Modifies dictionary words according to certain rules (append digits, uppercase, etc.).
- **Mask Attack**: Uses patterns (e.g., known length or character set).

7 Tools Overview: John the Ripper, Hashcat, fcrackzip, and More

A summary of popular password-cracking tools:

- `john` (John the Ripper): For various password hashes, widely used in forensic tasks.
- `hashcat`: GPU-accelerated password cracking, supports many hash types.
- `fcrackzip`: Specialized in cracking password-protected ZIP archives.
- `pdfcrack`: Used for PDF files requiring a password.
- `office2john.py`: Extracts hashes from Office documents for John to crack.

Each tool may support specific modes or file types, so consult the documentation.

8 Hands-On Examples

8.1 Cracking ZIP File Passwords

Using `fcrackzip` in a dictionary attack mode:

```
fcrackzip -v -D -p rockyou.txt protected.zip
```

Options:

- `-v`: verbose
- `-D`: dictionary-based
- `-p`: specify the wordlist file

8.2 Recovering Passwords from PDF Files

Two main steps:

1. Extract the hash:

   ```
   pdf2john.pl protected.pdf > hash.txt
   ```

2. Crack using John:

   ```
   john hash.txt --wordlist=rockyou.txt
   ```

8.3 Breaking Passwords of Word Documents

For protected .docx files:

1. Convert to a hash file:

   ```
   office2john.py protected.docx > hash.txt
   ```

2. Run John with a wordlist:

   ```
   john hash.txt --wordlist=rockyou.txt
   ```

9 Hash Extraction and Cracking with Hashcat

`hashcat` uses GPU acceleration for speed. Basic example:

```
hashcat -m [hash_mode] hash.txt rockyou.txt --force
```

The `-m` parameter (hash mode) is critical; for example, 9600 is often used for Office 2013+ encryption.

10 Best Practices, Legal Considerations, and Ethical Use

- Always ensure legal permission to crack or inspect secured files.
- Maintain detailed logs and chain of custody in forensic procedures.
- Verify that any password cracking is done ethically and for legitimate purposes (e.g., law enforcement, incident response).

11 Useful Resources and Wordlists for Practice

- Wordlist: `/usr/share/wordlists/rockyou.txt`
- John the Ripper
- Hashcat Wiki
- ExifTool

Conclusion

Digital File Analysis is vital for discovering hidden information, recovering lost data, and revealing potential evidence in forensic settings. By mastering tools like `binwalk`, `strings`, and `exiftool`, along with password cracking utilities such as `john` and `hashcat`, investigators can extract critical insights. However, always remain mindful of legal boundaries and ethical guidelines.