# MODULE 05 Defeating Anti-forensics Techniques

Lara Alofi

2110886

Y

Computer Hacking Forensic Investigator (CHFI)

Lab Session Identifiers

1. <https://labclient.labondemand.com/LabClient/e2e7e880-2440-40e3-b9ad-bbe98de96d56?showWhenStarting=1>
2. <https://labclient.labondemand.com/LabClient/c4bb1742-5048-4dbb-b85c-68b86a18a95b?showWhenStarting=1>

Username on EC-Council System

1. 2110886@uj.edu.sa

A screenshot of a computer

Description automatically generated

**Lab 01: SSD File Carving on a Windows File System**

In this lab, I learned how to perform SSD file carving on a Windows file system using Autopsy. The task involved examining TRIM-enabled and TRIM-disabled SSD images, which had been acquired as forensic evidence. The key focus was to understand how the TRIM functionality influences data recovery.

When TRIM was enabled, Autopsy was unable to carve files, demonstrating the challenge of data recovery when TRIM is active. However, when working with the TRIM-disabled SSD image, I successfully recovered carved files, specifically by expanding the CarvedFiles and Deleted Files nodes in Autopsy.

Through this process, I gained insight into how file carving can help retrieve deleted or fragmented files from unallocated space, which is crucial in digital forensics investigations involving SSDs.

**Lab 02: SSD File Carving on a Linux File System**

In Lab 2, I used Autopsy to perform SSD file carving on a Linux forensic image. After creating a case and adding the disk image Linux\_Evidence\_SSD.dd, I waited for the Carved Files folder to load. I successfully retrieved the carved file f0203152.jpg and extracted it for analysis. This lab enhanced my skills in using Autopsy for data recovery and reinforced my understanding of file carving techniques in a forensic setting.

**Lab 03: Recover Data from Lost/Deleted Disk Partition**  
In Lab 3, I used EaseUS Data Recovery Wizard to recover a deleted disk partition that contained malicious files from a victim's workstation. After logging into the Windows 11 virtual machine, I initiated a scan for the lost partition, which was identified as **Lost Partition-1.** Once the scan was completed, I was able to view and select the data within the partition for recovery. I directed the recovered files to a pre-created folder named **Recovered Partition**. The lab provided valuable insights into recovering lost partitions and highlighted the importance of such techniques in forensic investigations.

**Lab 04: Recover Data from a Partition that is Deleted and Merged into another Partition**

In Lab 4, I used R-Studio to recover data from a deleted partition that had been merged into another partition on a victim's workstation. After logging into the Windows 11 virtual machine, I merged the free space from the deleted partition into the existing Forensic Disk (F:), increasing its size. I then initiated a detailed scan of the F: drive using R-Studio to identify the merged partition, which was recognized as **Recognized1**. After selecting this partition, I began the recovery process, directing the recovered data to a designated folder on the C: drive. Despite the limitations of the demo version preventing the recovery of large files, I successfully retrieved the relevant data, showcasing how to handle complex forensic scenarios involving deleted and merged partitions.

**Lab 05: Extract password hashes from the target system using pwdump**  
In Lab 5, I extracted LM and NTLM password hashes from a Windows 11 machine using the pwdump7 tool. First, I copied the pwdump7 folder from the designated path to my Desktop for easy access. Then, I opened the Command Prompt with administrator privileges and navigated to the pwdump7 directory. By executing the `PwDump.exe` command, I successfully dumped the password hashes into the Command Prompt window. These hashes can now be utilized with various password-cracking tools to attempt to recover the admin account's password. This lab reinforced my understanding of hash extraction techniques crucial for forensic investigations.

The user ID of the Admin account extracted during this process will be noted in the answer field as required.

**Lab 06: Crack Application Passwords**  
In Lab 6, I learned to crack passwords for various protected files using forensic tools. I installed **Passware Kit Forensic** to recover the password for a Word document (`Confidential.docx`) and successfully retrieved the password. Next, I installed **Advanced Archive Password Recovery** to crack the password for a compressed file (`Confidential.rar`), configuring it for a brute-force attack. Finally, I set up **Advanced PDF Password Recovery** to tackle password-protected PDFs. Each tool effectively demonstrated the process of accessing sensitive information for forensic investigations.

**Lab 07: Detect Steganography**  
In Lab 7, I explored steganography, the technique of hiding information within other files to evade detection. I used several tools, including StegSpy to analyze a suspicious image file (`Model.png`) and identify the steganography technique used. After extracting hidden data from another image (`Sample.png`) using OpenStego , I successfully retrieved the concealed text file. Additionally, I used the zsteg tool in Ubuntu to examine another image (`cat.png`) for embedded information. Finally, I learned to extract hidden data from audio files with DeepSound . This lab enhanced my skills in detecting and analyzing files that conceal malicious content.

**Lab 08: Detect Alternate Data Streams**  
In Lab 8, I learned to detect and extract information hidden in Alternate Data Streams (ADS) using Windows PowerShell and third-party tools. The lab focused on how attackers utilize ADS to conceal malware and rootkits within seemingly innocent files on NTFS systems.

Initially, I disabled Real-Time Protection and used PowerShell to detect ADS with a specific command, which revealed files with hidden streams. I then opened two text files containing deceptive content to demonstrate how normal files can mask hidden data.

I extracted the hidden streams from these files using PowerShell commands, successfully displaying the concealed messages. Next, I installed NoVirusThanks Stream Detector to scan the same directory for ADS, allowing me to visualize the streams detected, including their names, filenames, and sizes. This lab enhanced my ability to uncover hidden information that could be crucial in forensic investigations.

**Lab 09: Detect File Extension Mismatch**  
In Lab 9, I used Autopsy to detect file extension mismatches, a technique used by attackers to disguise malicious files. After creating a new case and analyzing the Windows\_Evidence\_001.dd image file, I identified files with altered extensions. For instance, the file MultiplePages-Fixed.pdf had its MIME type revealing its true nature, showcasing how attackers can hide malicious content by changing file extensions. This lab reinforced the importance of recognizing such tactics in forensic investigations.

**Lab 10: Unpack Program Packers**  
In Lab 10, I learned how to unpack malicious program files disguised by attackers using program packers. I started by identifying the packer for the Infected.exe file using Detect it Easy (DiE) and ExeInfo PE , which revealed that it was packed with UPX . I then utilized the UPX tool to unpack the file via PowerShell, resulting in a new file, Unpacked.exe , that contained the original executable. This lab highlighted the importance of recognizing and unpacking packed files in forensic investigations to uncover hidden threats.