Unit 1 – Introduction to Digital Forensics

Unit 1.1 – What is Digital Forensics?

Broadly speaking, **digital forensics** is the examination of data stored in computer systems in order to discover **digital artifacts**. In a law enforcement context these artifacts may be called **digital evidence**.

The computer system examined may be a personal computer, a corporate computer, a tablet, a mobile phone, a USB (or other) portable drive or even a game system. New devices emerge all the time. Consider the smart watches and activity and sleep tracking devices. In a recent case the time of death of a murder victim was established using the data from a fitness tracker. If the device has any kind of storage in it, then it falls under the "computer system" umbrella. By this definition the computer system in a car would qualify. The computer in a microwave oven wouldn't qualify unless it stored a history of its activity.

Digital artifacts are artifacts that are collected from any device that stores data, usually with some purpose in mind. The most common venues are law enforcement and corporate crime, but there are others as well.

In many ways digital evidence is like fingerprints. Fingerprints in a traditional forensics investigation can establish the fact that a person was present in a place at some time in the past. Similarly, when you use a computer system there are **digital artifacts** stored. Unlike fingerprints digital artifacts often have **timestamps** which can be used to establish a point in time when some activity was undertaken (but be careful, timestamps can be tampered with!).

A classic example of digital artifacts is an **Internet browsing history**. In recent years there was a court case where a man was on trial for murder after his child died in a hot car. During the trial the prosecution introduced evidence from his browser history that he had used a search engine to find out how long it would take for an animal to die in a hot car. This helped established premeditation which changed the nature of the criminal charges.

Your browsing history may be recoverable even if you clear it. We will see later that deleted files can be recovered if you know how.

Other artifacts are stored in **registry keys** or other databases. If you attach a USB portable drive to a Windows computer it stores the serial number of the device in the Window's registry. It is very easy to establish that a particular portable drive was connected to a particular computer at some point in the past. You just need to know where to find the id number on the device and in the registry.

Another classic example of digital artifacts is **metadata**. Certain programs store data that is not readily visible to the user. The type of metadata depends on the program. An image file often will store the model number of the device from which the image originated (a camera for example) or the program it

was created on (photoshop for example).

The BTK killer eluded capture for decades but couldn't resist communicating with law enforcement. At one point he sent the police a floppy disk with a deleted RTF (rich text format) document on it. Apparently he wasn't aware that inside an RTF document the computer name and user name are stored, which made it a simple matter to find him.

Sometimes digital forensics is used to defend a person that is accused of a crime. Defense attorneys hire digital forensics practitioners as “expert witnesses” to try to clear their client or establish a reasonable, lawful explanation for their activities.

Digital forensics is also used in a corporate setting. Obvious examples are theft and distribution of intellectual property (movies, music and trade secrets for example) or credit card numbers. Other examples are responses to break-in attempts or incidents, or evidence of employee misconduct.

The tools used in digital forensics can also be used to conduct research. When a new computer virus is suspected, researchers can compare **images** (file copies) of a file system before and after the introduction of the virus. It is a simple matter to find the location of all changes made to the system. Once the locations of the corrupted files are discovered they can come up with a plan to identify and mitigate the virus.

Unit 1.2 – Some Thoughts about Course Expectations

From all the examples given in the previous section one thing should be clear: a digital forensics investigator must have knowledge and skills that go well beyond that of the average computer user.

That being said, as the instructor for this course I believe that if you can pass MA112 or MA116 then you can learn number systems, representation of alphabets and how to use digital forensics tools.

There may be many new things that I will ask you to learn in this course. At first they look complex, but they really aren't. If you can stay with the ideas long enough you will see that they aren't as hard as they look at first.

The ideas in the first part of this course (hexadecimal, ASCII, Unicode, etc) are essential to your success later in the course. Please take the time to learn these things well. If I'm lecturing and you need to stop and review an earlier topic then please raise your hand and ask me to do so. I will appreciate the feedback.

As a final note, the students in this course are a mixture of CIS and non-CIS majors (this course is crosslisted as both CM203 and CJ290, so make sure you are enrolled in the course that will help you the most with your chosen degree).

The CIS majors have seen much of the basic material many times before. It is unlikely that any non-

CIS majors have seen binary, hexadecimal or Unicode before.

I am committed to the idea that all students be treated equally in this course. I will teach it as if no one in the course has seen any of this before and only ask you to demonstrate knowledge that I have covered in this course.

To the CIS majors, please be patient when I'm going over things you already know. And just because you've seen some of it before that doesn't mean that you can skip class. Every day there will be something you haven't seen in your other CM courses.

To the non-CIS majors, please don't let the fact that the CIS majors know some of these things already discourage you. We will get to stuff they don't know soon enough. Your grade is based on what you can demonstrate on the tests and projects. Trust me that the expectations will be reasonable and if you do all the work conscientiously you will get a good grade.

Most of the projects will be done during class time. There will be very little homework to do outside of class. Thus it is very important to not miss any classes. However, life happens and it is possible that you miss a class occasionally. Please see me as soon as possible after a missed class and we'll figure out a way for you to make up the work. It is up to you to take initiative to do this. I'm not going to track you down.

Unit 1.3 – Types of Storage

There are many ways that **data** are stored on a computer system (the word “data” is plural, by the way, the singular is “datum”).

The good news for us is that we can look at all storage systems the same way. We don't need to be concerned with the physics of how the bits are actually written to and retrieved from the media.

Any storage device is basically a collection of bits of data. The data are stored in “chunks” (for lack of a better word at this point) and each chunk has an address. The addresses start at zero and go up to some (usually large) number. How many bits are in each chunk depends on the device and formatting.

Computer storage comes in two general categories: **volatile** and **non-volatile**.

Volatile memory is cleared of all data when the power is shut off. Non-volatile memory retains its data even when the power is turned off.

So what good is volatile memory if it can't store data between power-ups? The answer is that volatile memory is much faster than non-volatile memory, and is used as a kind of scratch space for data while it's being manipulated. For example when you type a term paper into a word processor the letters you type are stored in volatile memory until they are saved in a file, at which point they are stored in non-volatile memory.

The volatile scratch space referred to in the previous paragraph is called **Random Access Memory** or **RAM** for short. It is a core component of every computer system. As the name implies, any part of it can be accessed at any time.

In order for a computer program to run it must be loaded into RAM first. When a data file is manipulated it must be loaded into RAM first. RAM is the place where changes are made to the data.

In a nutshell:

RAM: volatile, fast, expensive, small

Other Storage: non-volatile, slow, cheap, large

For simplicity we will refer to non-volatile storage with the term "**drive**".

Today's standard for non-volatile storage is the **hard disk drive**. Inside is one or more disk platters that rotate under read/write heads.

The newer drives are **solid-state drives**. They don't have a disk inside them. Thus they shouldn't be called "disk drives". It is likely that solid-state drives will replace hard disk drives within the next decade or so.

Then there are the ubiquitous **flash drives**, also known as thumb drives, jump drives, etc.

For our purposes all these different types of storage devices can be treated generically as "drives".

The basic unit of storage (chunk) is the **sector**. Sectors nearly always have a size of 512 bytes (4096 bits). Sector numbers start at zero and go up to some large number.

Unit 1.4 – Operating Systems

The **operating system** (**OS** for short) is the program that runs when you turn on a computer. It allows the user to start and stop programs and it manages the shared resources of the computer.

There are many different operating systems out there, but we will only look at a few of the more popular ones. Clearly there is more bang for the buck to spend your time learning the operating systems that most people use first.

These are for a PC/Laptop (in order of popularity):

Windows
Mac OS X
Linux/UNIX

and for mobile devices:

Android
iPhone OS

We will spend most of our time working with Windows since it is the most popular OS today.

Linux has the advantage of being 100% free and **open-source**. Open-source means that the source code of the programs is freely available and it is probably supported by volunteers that might not receive any compensation for their work (though they will often accept donations).

The main difference between Linux and **UNIX** has to do with history. The first version of UNIX was written at AT&T Bell Labs in the early 1970's. It was given away to universities in the hopes that they would develop it, which they did. Later on, Bell labs tried to enforce their patent rights and the legal rumblings began. In the early 1990's Linus Torvalds wrote Linux from scratch to emulate UNIX. It has none of the original Bell Labs code in it, so it is free from legal wrangling.

Most of the time we will use Linux in this class because there are tools built-in to Linux that are very useful for forensic analysis. The same tools can be added to in Windows but often they are not free. University professors like free stuff because it helps stretch out the department budget.

There can be some confusion in forensics regarding the use of the term “operating system”. When doing a forensic examination there is the system that the examiner uses to run the forensics tools, and the system that is being examined by the examiner. We will use the term **forensic host** (or just host) for the machine that is used to run the tools and **forensic subject** (or just subject) to be the machine that is being examined. The host and the subject don't need to have the same OS.

Most of the time in this course the host will be Linux and subject will be Windows.

I should probably mention that a given computer system can usually run any number of operating systems. The computers in MO51 are set up to multi-boot to one of two different operating systems. The first choice is Linux Mint. The second is Windows 10. There’s a switch on the front of the computers that can switch between the operating systems. Please make sure the power is off before flipping the switch.

Beyond this, you can boot to an OS that is stored on a CD or DVD or even a USB drive.

If you have the ability to boot a computer to an OS stored on a CD/DVD or USB drive then any security set up on the system (with the exception of encrypted partitions) is ignored. We will see later that any file on the system is accessible using the alternative OS.

This lack of security will probably spur the use of encrypted partitions in future computer systems. That scenario is a digital forensics examiner's worst nightmare.

On any of the PC operating systems above you can run just about any other OS in **Virtual Box**. Virtual box is a software package that allows multiple guest operating systems to run on a host operating system.

Unit 1.5 – File Systems

Drives are devices that can store large amounts of data. We learned before that drives are divided into sectors. Some files are large and require many sectors to store them. But how do we know which sectors a file is contained in? And in what order?

A **file system** is a way to organize the sectors on a drive so that we can find the files that we want quickly.

File systems are organized into **directories**. A directory is a place where we can store files and other directories. It is a structure known as a **tree**. The starting point is called the **root** directory (kind of like a tree has a root, but in this case the root is at the top of the tree). The root directory has subdirectories and each of those can have subdirectories much like the branches of a tree.

Most file systems also have some way to indicate **ownership of directories** and files and a way to indicate the **access permissions** (who can read them, change them, etc) for that particular directory or file.

Over the years there have been many schemes for organizing sectors into file systems. We will learn about the most popular ones because these are the ones we're most likely to see in a forensic investigation.

Here's a brief list:

FAT (FAT12, FAT16, FAT32)
NTFS
Ext (Ext2, Ext3 & Ext4)
HFS
ISO
UDF

(There are also file systems for mobile devices, but this goes beyond the scope of this course.)

FAT and **NTFS** were developed by Microsoft for their DOS and Window's systems. The **Ext** systems were developed for Linux systems and **HFS** was developed by Apple for their Macintosh systems.

ISO and **UDF** are used for CD, DVD and BluRay data disks.

Of all these NTFS is probably the most common today, although FAT is a close second as it is still used on most USB drives.

We will dig in later and look at how FAT and NTFS file systems work. I won't lie to you, they are complicated. But we'll take our time and go one step at a time and we'll understand the basics of how the OS can navigate from directory to subdirectory and find any file in a file system.

While we're at it, we will see where timestamps and file ownership and permissions are stored. Forensics tools are able to show you **everything** in the file system in its raw form. Unless it's encrypted, if it's there, you can see it.

A big advantage of looking at files in raw form is that you can see the unallocated space at the end of the data stored in a file. Files are stored in fixed-sized blocks. Usually the last block is not completely full and has what is called **slack space** at the end. Sometimes people hide data in slack space, and you can easily see that data with forensics tools.

Then there's the matter of **deleted directories and files**. Because of the way data is stored in a file system it may be possible to see directories and files that have been deleted. Again, forensics tools allow access to everything in the file system, including items that have been deleted.

In the case of a deleted directory sometimes the directory listing is rewritten. Sometimes the directory is simply marked as deleted with no other changes. This really depends on the file system.

In the case of deleted files, the file blocks usually remain unchanged on the drive, but the directory entry that points to those blocks is removed or marked as deleted. Unless the user uses a program to do a "**secure delete**" the file blocks remain unchanged. With secure delete the file blocks are actually zeroed out. You can see that if a file hasn't been deleted for very long, and has not been zeroed out, it may be recoverable partly or in its entirety.

There are tools to recover all deleted files in a file system provided they haven't been written over. You will see later that they are very effective.

It is worth mentioning that a drive with one file system on it can be overwritten to create another file system. This is called "**reformatting**". To save time the drive is not normally zeroed out, so some data from the previous file system will still be there. You should be aware that much of what you see in a forensics investigation will be junk left over from earlier activity. It's your job to figure out what is junk and what isn't.

Unit 1.6 – Legal Aspects of Digital Forensics Investigations

This course focuses primarily on the technical aspects of digital forensics. It is also important to point out that digital forensics investigators also need to learn and follow legal guidelines in their daily work.

I'll summarize only the most basic guidelines here. If you go on to specialize in digital forensics investigation you will learn the rest along the way.

1. At every step in an investigation you must **record case information** along with a description of what activity was performed and what results were obtained.

2. Part of the documentation required is to establish **chain of custody** for each piece of physical evidence. Chain of custody records who possessed each piece of evidence at any point in time from the beginning of the investigation to the end of all legal proceedings (and beyond).

3. You will make an **image** (a data file that contains everything on the drive) of each drive that you examine. You will use a **write-blocker** to ensure that nothing on the drive is changed during your investigation. Your investigation will actually be done on the image file.

4. Your documentation must be neatly organized in such a way that you can find things quickly if you are questioned about the details.

5. You may have to testify in court regarding your findings. Many cases are ended with a plea agreement, but you must be prepared for any case to go to trial.

If you take **CM303** you will learn much more about the legal aspects of digital forensics (most likely from a KBI or FBI digital investigator).

Unit 1.7 – What We Will Do in this Course

First, we'll study how to use Linux on the **command line**. This is important since many useful tools are available only on the command line.

Next we'll study of how numbers are stored in computers. They are stored in **binary**. So we will learn to count in binary and convert binary numbers to decimal and vice-versa.

We'll see how binary numbers can be represented **hexadecimal** (hex for short). Hex is the bread and butter of forensics, and it isn't hard to learn. You will be given a table that translates between hexadecimal, binary and decimal. Think of it as the Rosetta Stone for hex.

Then we'll look at how English letters (basically the keys of a typewriter) are stored in hex. This is very important since it allows you to tell whether a given piece of data stored on a drive is storing a

number or words in a message. You'll be given a table that allows you to translate between hex and letters (but most forensics tools do that automatically).

After that we'll look at how letters of all languages of the world are stored in **Unicode**. The tables for this system would be thousands of pages long, so I'll just show you where to find them on a website.

Once you know how the numbers can be interpreted in human terms, we'll learn how to look at **hex dumps**. A hex dump is a raw listing of some part of a drive in hexadecimal (most tools also show you typewriter letters as well, though binary data looks like gibberish).

Once you have been trained to read a hex dump, our real study of file system forensics can begin.

We'll spend some time looking at hex dumps of files. There are ways of determining the type of a file by the **metadata** in the file, even if the file extension has been lost or changed. We will examine what type of metadata can be recovered from various types of files.

We will see that some files are readable in their raw form and some are compressed in such a way that they are not readable. Even these files often have metadata that you can read.

There will be a survey of digital forensics tools and advantages/disadvantages of each.

Next we'll look at **Window's artifacts** (Internet browser history/cache and the registry). There are many useful tools for recovering Window's artifacts.

We'll learn how to make **images** of drives. This is a time-consuming process so we'll do some other stuff while the data are copied. We'll learn how to prove that a copy of a drive is a perfect copy.

We will learn about **partition tables** which tell the OS where various partitions are stored on a drive.

We'll dig in to see how **FAT file systems** work. We will use a tool that shows clearly how to get to the root directory and to navigate through the entire file system

We'll investigate **NTFS file systems**, and see that it is much more complex than FAT file systems. Like any new thing over time it will not seem as daunting.

If we have time we'll look at other file systems (Ext, HFS, ISO, UDF).

If we still have time there will be other miscellaneous topics such as **steganography**, **encryption**, **cryptanalysis** and **password cracking**.