Abstract: In this paper, we describe forensic live analysis and event reconstruction methods in digital crime investigation. This information is forensically interesting because it helps to determine the origin of events by gathering data for analysis and applying the methods of event reconstruction for evidential purposes in the court of law. Our investigation is focused on Linux systems. We have noted the effectiveness of existing automated event reconstruction systems and we present an experimental study that describes the forensic live response and event reconstruction in digital crime investigation.

I. INTRODUCTION

1.1. Background

Currently, most of the attention in digital investigations has focused on the search for and collection of digital evidence[1][2]. The digital forensic tools on the market preserve the state of a system or examine a system to find evidence, and after every incident, the universally asked questions are ‘what happened?’ and ‘how did it happen?’[3]. But, rarely is the question asked ‘how has the data been collected?’; ‘how has it been interpreted?’; ‘how has the resulting interpretation been conveyed to its audience?’ and ‘why an object may be evidence?’ Collecting an object and examining its properties is interesting, but for the evidence to be useful; we must identify, investigate and reconstruct ‘what caused that object to have those properties?’; ‘what evidence can be obtained for presentation in the court of law?’ Event reconstruction methods or event analysis tools are required to examine evidence and to identify why that evidence has particular characteristics. As discovered from past research many events occur at a crime scene, including ones that occurred prior to the incident, and they must be understood so that the incident can be fully explored.

1.2. Forensic Live Response

Forensic Live Response is a process of recovering and safely storing volatile data for later analysis; it gives the forensic examiner the chance to collect volatile evidence in a ‘human-readable’ format that is easier to peruse than when it is stored as a binary version [4]. In today’s digital investigation, Forensic Live response is found to be vital because when the system is shutdown, evidence in volatile memory might be lost [4]. This evidence might be useful in analysing what has occurred at the crime scene and reconstructing events leading up to the crime [5].

II. LINUX: VOLATILE MEMORY

The approach of volatile memory image analysis is similar to live response or live acquisition in that an investigator would first establish a trusted command shell and, then, they would establish a data collection system and a method for transmitting the data. Due to the volatile nature of running memory, the imaging process is taking a snapshot of a ‘moving target’. This leads to difficulties in reconstructing events accurately and therefore it makes this research topic an area of interest.
2.1. Volatile memory analysis

We have examined that the purpose of live response is to collect all relevant evidence from the system that will likely be used to confirm whether the incident occurred. This is what our research focuses on. Furthermore, while the implementation of the live response process is important, there are significant setbacks, including the following:

1. Live response is considered not to be repeatable. The information in memory is volatile and changes as the system is used [6].
2. After live response process, investigators cannot ask new questions later because the live response process does not support examination of the evidence in a new way. This is mainly because the same inputs to the tools from the collection phase cannot be reproduced [7]. This research project may mitigate this effect by enabling reconstruction of the captured event to occur. By the analysis phase, it becomes impossible to learn anything new about the compromise and once critical evidence is missed during collection, it can never be recovered again.

2.2. Non-volatile memory analysis

On the other hand, non-volatile memory analysis shows promise in that the only source of evidence is the physical data and collection of this has become more commonly practiced [8]. Thus, we have examined non-volatile analysis and arrived at these differences with the volatile analysis:

- Non-volatile analysis limits the impact to the compromised system by not require interaction with it. However volatile analysis may involve interacting with the compromised system if firewire memory capture, or similar, is not used. Thus volatile memory capture may affect the evidence being capture by overwriting important blocks of data [9].
- The nature of non-volatile data collection supports asking new questions later. Contrary to volatile memory collection, where collecting data from the memory at a different point may result in slight chances to data being collected. This means that knowledge is required regarding how data in memory of a running system will alter over a period of time to ensure that the analysis conducted is sound [10].

2.3. Drawback of volatile memory analysis

One of the greatest drawbacks with volatile memory analysis is that the tools’ support is not mature enough [11]. This is because with every release of a new operating system, the memory structure changes. However, development of memory analysis tools has been gaining popularity recently. This is an emerging field and new ground is being broken across the area of study [12].

III. EVENT RECONSTRUCTION

3.1. Event reconstruction definition

Collectively, the process of identifying the underlying conditions and reconstructing the sequence of events that led to a security incident is referred to as ‘event reconstruction’. Typically, event reconstruction involves forensic investigators manually sifting through evidence and making various hypotheses about the possible sequences of events that could have led to that security incident. The degree of difficulty and the accuracy of the results of the reconstruction process differ from case to case.

3.2. Reconstruction process

The reconstruction process in cases of incidents is commonly referred to as ‘operational forensic analysis’ or intrusion analysis while prosecutorial forensic analysis often
leverages the presence of additional evidence in the form of audit logs [12].

3.3 Automated reconstruction systems

In the past few years, many researchers have developed automated reconstruction systems that rely on a priori audit logging to help the reconstruction of events of incidents. Examples include Backtracker [12], Process labels scheme [13] and the Forensix [14]. The key idea is that with more information logged about the events during the normal operation of a system, reconstruction becomes easier and can be automated for investigation [1].

3.4. Effectiveness of reconstruction systems

Despite the growing body of literature regarding automated reconstruction systems, there is still very little work that quantifies their effectiveness. Therefore a rigorous study that quantifies their effectiveness is essential for the following reasons:

1. Event reconstruction systems often provide multiple hypotheses regarding the possible causes of a security incident. If false-positive rates are available, they can be used as priors for calculating the likelihood of each hypothesis, allowing investigators to order or prioritize the different hypotheses [15].

2. Systems administrators and forensic investigators need guidance as to which reconstruction systems to deploy for their particular framework and circumstances. This is because reconstruction systems that are suitable for certain situations, misbehave in others [16].

3. Researchers can use such a study as a guide towards identifying the challenges that need to be tackled in order to build better reconstruction systems [17].

4.1. Experimental studies

We have conducted an experimental study that evaluates the forensic live response on Linux systems and event reconstruction methods of digital investigation. We have noted that a digital object is considered to have a characteristic or unique feature, based on their creator and function. For example, the characteristics of a hard disk sector will be different when it is used to store the contents of an ASCII text document versus a JPEG image. We can use these characteristics to identify the data because, it is assumed, that the state of an object is a value of its characteristics. If a letter were changed in an ASCII text document, then the object corresponding to the file would have a new state. Similarly the state of a running computer process changes every time data is written to its memory because a digital event is an occurrence that changes the state of one or more digital objects. If the state of an object changes as a result of an event, then that is an effect of that event.

Some types of objects have the ability to cause events and they are called “causes”. We have observed that because digital objects are stored in a physical form, then their state can be changed by both physical and digital events and the object is considered evidence of an event if the event changes the object’s state. This means that the object can be examined for information about the event that occurred. However, future events could cause an object to no longer have information about past events. Therefore, every object is evidence of at least one event, because there had to be an event that created the object.

Some environments have developed policies that prevent certain events from occurring whereas, an incident is an event or sequence of events that violate a policy and more specifically, a crime is an event or sequence or events that violate a law. In particular, a digital incident is one or more digital events that violate a policy. In response to an incident or crime, an investigation may begin. An investigation is a
process that develops and tests hypotheses to answer questions about events that occurred. For example, “what caused the incident to occur”, “when did the incident occur”, “where did the incident occur” and “why an object may be evidence”. Our contributions are the following:

- Using our approach, we provide data describing the Forensic live response of data acquisition and data analysis on Linux systems.
- We have commenced further investigation on the techniques and process of event reconstruction of data analysis on Linux systems.

From the memory captured on the Linux system, we can reconstruct the events that had previously occurred for investigation purposes and for evidential use in the court of law [1]. With the case of volatile data, we have identified that once volatile data have been gathered and have moved into persistent data, it is compulsory to validate the images. We have also noted that volatile data can be correlated and compared with shell histories, localhost logs and network logs.

4.2. Experimental Study: Forensic Live Response

Our systematic approach to Forensic Live Response techniques build on previous work by [18]. We have prepared live response materials for event reconstruction based on the first challenge in measuring the effectiveness of reconstruction techniques to decide upon a live data analysis. The idea behind this is that the effectiveness of the reconstruction process is directly dependent on the accuracy with which causal relationships between events are inferred.

4.3. Experimental Study: Event Reconstruction

In Fig. 1, the reconstruction phase identifies an event (e₁) for which evidence exists to support its occurrence. Consequently, the reconstruction adds an additional dimension to that evidence, instead of having information about only the final state of an object; information is gained about the preceding states, as shown in Fig. 1, where X, Y and Z represent three different characteristics of evidence gained after an occurrence of event. These characteristics can be used to recreate all possible ways that the event could have been caused. The diagram illustrates the reconstruction attempts (R) to deduce the previous states by examining the events in which an object may have been involved. The current focus has been on the recognition and identification of digital evidence using an object’s characteristics [19]. Generally, we will focus our research on the investigatory process of forensic live response and the event reconstruction methods in digital crime investigation.

![Fig. 1. Illustration diagram of reconstructing events](image)

5.1. Observation

Looking at the different methods of identifying data from the memory dumps, the general trend is that the less information we want to link to the analysis stage for example, process information or file name, the more stages can be identified, but not linked to other information in the memory dumps. By identifying the suspicious data in memory dumps and linking this information to process structures, we obtain information about the origin and usage of this suspicious data and this will reduce the amount of unknown data in memory dumps.

5.2. Findings

The observation above suggests another important area of our research as referred to Fig. 1, the cause and effect
correlation of event reconstruction. Cause and effect correlation is when there is a change to the characteristics of an effect object that must have occurred because of a characteristic from a cause object. These include roles and methods which are unique to each event. As discussed above, this area is emerging as a promising approach to crime investigation for digital and evidential purposes in the court of law.

5.3. Research Presentation

Based on our findings, we intend to expand the existing research in correlating the cause and effect of events by integrating a role-based event reconstruction model about why evidence exists and how automated executable analysis investigation tools can be applied. While more defendants claim that evidence was planted on their systems, the investigators will need to identify if the defendant is actually identified as the cause of events i.e. download a file or it was planted there by someone else [7]. This abstract model will help to perform the reconstruction tasks. It will allow a law enforcement laboratory to identify the application’s process that can initiate events. Knowing the state of a system is required to show the reliability of the evidence, but, it is difficult to know what state a closed source application must be for it to perform an event.

VI. CONCLUSIONS

In this paper, we have demonstrated the forensic live response approach and event reconstruction methods of digital investigation in Linux systems. We have illustrated, based on our approach to live acquisition as shown in Fig. 1, and we believe that the changes to the effect object are related to those of the cause object. In Fig. 1, we have proven a characteristic effect of reconstruction and we propose that the location of the effect object is relative to the location of the cause object. As we can use its location to identify the location of the cause, then, that location may contain additional evidence. Perhaps, our primary work will be if an event is in progress during the evidence collection, then the time and the cause of the event may be determined using the current state of the event via the rules and laws of that event. The on-going research on forensic live response and event reconstruction becomes important because investigators must be able to defend their hypotheses about why an evidence exists [20]. We are not aware of anyone that has attempted to address all of these problems: incident response and live analysis; methods for interrupting the execution for live acquisition; methods for performing live analysis on systems without interrupting the execution sequence; methods in relation to the cause and effect of event reconstruction; abstract model of why an evidence exists; and the automated executable investigation analysis tools in event reconstruction of digital crime investigations to generate ‘data fact’.

REFERENCES