Digital Evidence Composition in Fraud Detection

Sriram Raghavan[1] and S.V. Raghavan[2]

[1] Information Security Institute, Queensland University of Technology, Brisbane 4001, Australia
sriram.raghavan@student.qut.edu.au
[2] Network Systems Laboratory, Dept. of Computer Science & Engg., IIT Madras, Chennai, India
svr@cs.iitm.ernet.in

Abstract. In recent times, digital evidence has found its way into several digital devices. The storage capacity in these devices is also growing exponentially. When investigators come across such devices during a digital investigation, it may take several man-hours to completely analyze the contents. To date, there has been little achieved in the zone that attempts to bring together different evidence sources and attempt to correlate the events they record. In this paper, we present an evidence composition model based on the time of occurrence of such events. The time interval between events promises to reveal many key associations across events, especially when on multiple sources. The time interval is then used as a parameter to a correlation function which determines quantitatively the extent of correlation between the events. The approach has been demonstrated on a network capture sequence involving phishing of a bank website. The model is scalable to an arbitrary set of evidence sources and preliminary results indicate that the approach has tremendous potential in determining correlations on vast repositories of case data.

Keywords: Evidence source, Event, Correlation function, Probability function.

1 Introduction

In a digital investigation, investigators deal with acquiring digital data for examination. Digital records vary in forms and types. Documents on a computer, telephone contact list, list of all phone calls made, trace of signal strengths from base station of a mobile phone, recorded voice and video files, email conversations, network traffic patterns and virus intrusions and detections are all examples of different types of digital records. Besides, a variety of new digital devices are being introduced with rapid advances in digital technology which are capable of storing such digital records. Coping with such advances has become challenging owing to the use of proprietary data structures and protocols in most devices rendering them difficult for interpretation without relevant documentation, let alone, in a forensically sound manner. The large volumes of data collected in typical cases can be attributed to this variety and sifting through them can be enormously time consuming. Yet, it is important to quickly sift through these large volumes of data and deal only with the relevant material. However, even this could impose a significant challenge. It then becomes the

S. Goel (Ed.): ICDF2C 2009, LNICST 31, pp. 1–8, 2010.
© Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering 2010

duty of the investigator to determine which entities are connected and in what manner. From a forensic standpoint, there is too much entropy in the forensic examination process to capture all data and process it manually. This is an enormous challenge facing investigators. Irrespective of these challenges, all records must be examined after acquisition in a uniform manner and the investigator needs to determine the events contained within these records which may have contributed to the case at hand. There is a need for integrating and analyzing information from such disparate sources.

Hosmer [6] calls for the need to standardize the concept of digital evidence to provide a common platform for investigators to perform forensic analysis. Drawing parallel from physical evidence acquisition process, he suggests adopting a methodology that is similar to how physical evidence are stored and organized. However, since digital evidences can be altered, copied or erased, he proposes the 4-point principles of authentication, integrity, access control and non-repudiation while handing digital evidence. Cohen [4] describes the PyFlag network forensic architecture, which is an open-source effort in providing a common framework for integrating forensic analysis from diverse digital sources. However, Pyflag does not attempt to identify correlations at the application level, which is fundamental to forensic analysis once the integrity of the data is established. In the context of the investigation, it is essential to analyze the data contained in these sources uniformly, irrespective of semantics and storage formats. Case et al [3] propose the *FACE* framework for performing automatic correlations in forensic investigation. However, the framework is structured to only consider static and known relations in data (for example, linking network socket in memory to TCP requests in packet capture) especially when significant case detail is available a priori. Raghavan et al. [8] propose the FIA framework as a platform to perform unified analysis at the application level. Our paper explores that territory to good effect by demonstrating the model on a fraud detection case to determine correlations on arbitrary pairs of events across different sources using time of occurrence of events. The rest of the paper is organized as follows. In Section 2, we review recent work reported in literature in digital forensic analysis. In Section 3, we present our evidence composition model and describe its implications to determining correlated events across evidence sources. In section 4, we apply our model to the fraud detection case and present our observations. In section 5, we make inferences based on our observations to determining correlated events across multiple sources. In Section 6, we conclude with a brief summary of the work done and propose directions for future work.

2 Recent Work

Gladyshev and Patel [5] propose a finite state model approach for event reconstruction. They demonstrate that even a simple printer investigation problem can have exponential state space for analysis. In the context of current cases, clearly such a system is impractical and newer methods are needed to simplify the state space analysis. Carrier and Spafford [2] propose a method for analysis using the computer history model. However, like in the finite state model case, the application is not practical to current case complexities. Jeyaraman and Atallah [7] present an empirical study of automatic reconstruction systems. Their paper examines different systems using an intrusion case. However, unless events are clearly defined a priori it is generally

difficult to identify and determine these events which render the process of little use. Bogen and Dampier [1] propose a case domain modeling approach for large scale investigations and define case specific ontology using UML. Wang and Daniels [9] propose an evidence graph approach to network forensic analysis and build a correlation graph using network captures. However, both approaches describes above require significant modifications before they may be adopted into another investigation setup. Such modifications are often very time consuming and unwarranted. As a consequence, these works has been of little use in practical forensics and more research is needed to bridge this gap.

In summary, there has been consensus on the fact that it isn't easy to quantify the value of digital evidence and hence measure the relative value of recorded events from the same or across evidence sources. With the growing size of digital evidence repositories and the advancements in technology, it is humanly impossible to match speeds with manual forensic examination and analysis. Newer methods and approaches are essential which explore the domain of integrating recorded events at the application level and provide scope for automation sometime down the line.

3 Evidence Composition Model

Consider an arbitrary collection of evidence sources under $E = \{E_1, E_2, E_3, \ldots E_n\}$. For simplicity, let us assume that each source is a homogeneous collection of evidence under the context of a single case. For example, E_1 could refer to a collection of Microsoft Office documents obtained under the NTFS partition of a hard disk, E_2 could refer to all emails, associated file attachments and business contacts' names acquired from the OST archive of the Microsoft Outlook mail client, E_3 could refer to all log file entries on a web server, and so on. In effect, each source can be uniquely identified and its contents searched in a uniform manner without having to concern one regarding intermediate forensic processes.

From this collection, let us pick two events e_i and e_j from respective sources E_i and E_j. Let t_i and t_j represent their respective time of occurrences on the real time clock. In the sample collection listed above, e_i could indicate creation/access times of a particular file and e_j could refer to the time when an email was sent from the Outlook client. In a digital investigation which involves reasoning with the occurrence of certain events and in some cases the relative times of occurrence, the interval between two events could often hold the key to providing valuable insights into the case, if not help solve it. In this example, the difference in times of occurrence $t_j - t_i$ (without loss of generality, we assume that $t_j > t_i$ on the real time clock) becomes an interesting parameter to monitor.

$$\text{Let, } \Delta t = t_j - t_i$$

Since the relative times and time intervals become crucial to the case, we define two thresholds δ and Δ on the time interval as below:

$$\text{If } \Delta t > \Delta, \text{ then the events are uncorrelated;}$$

$$\text{If } \Delta t > \delta; \text{ and } \Delta t < \Delta, \text{ the events are moderately correlated}$$

$$\text{If } \Delta t < \delta, \text{ then the events are strongly correlated}$$

Now the actual values that Δ and δ can take will be decided based on case specifics and often on the types of sources which the events e_i and e_j belong to. Suppose we assume that the values for Δ and δ are given to us by forensic experts, we can build a correlation function over arbitrary pairs of evidence sources (E_i, E_j). This function f: $\mathbf{R} \rightarrow \mathbf{R}$ can map arbitrary set of time event occurrences on the real line over the range [0, 1). Such a function can be as trivial as a linear real function or as complex as combinations of non-linear expressions mapped on the range of f. However, finding such a mapping which can accurately account for varying levels of time difference intervals, even within the same pair of evidence source is a challenge. Domain heuristics is expected to provide simple effective solutions in this regard.

3.1 Search Problem

Note that once the values of the thresholds Δ and δ are assigned, the problem of identifying pairs of correlated events boils down to a search problem on the time interval space. Depending on the requirements of a particular case, the problem is cast as searching for pairs of events (e_i, e_j) such that their occurrence interval $t_j - t_i$ is separated by no more than δ. Having determined such pairs the investigators can then proceed to drill down to the details of such pairs according to case requirements.

3.2 Complexity Analysis

On any arbitrary pair of evidence sources, the search problem amounts to identifying a particular event from a list of events recorded on a source and then determining another similar event on a different source for computing the time interval. Without loss of generality, this can also be performed on the same source which may provide additional computational benefits. The naïve approach suggests that the time complexity for a sequence of N recorded events on any source is $O(N^2)$. However, it is not unreasonable to assume that the sequences of events reported on the sources are intrinsically time ordered. This implies that having determined one event on one of the sources E_i (say), with an O(N) search, it is sufficient to compute the relative position of this event on the other source E_j. Since E_j is time sorted, we adopt the binary search algorithm with complexity O(log N) and the overall search complexity reduces to O(N log N). Further, if forensic experts can advise on specific time intervals within which such events could be analyzed in addition to specifying the values of Δ and δ, the complexity could be further lowered based on this information.

The authors acknowledge the fact that merely providing a correlation function based on probabilities does not suffice in a court of law. It is integral to the process of forensics to establish the events that occurred and their relative sequences beyond the realm of doubt. However, the concept of correlation does allow one to identify pairs/sequences of time ordered events with special relevance to the case at hand. It is then analytically possible to lower thresholds and empirically determine the lower bounds on Δ and δ where correlation becomes meaningful in a given context. In the next section we apply the model to a hypothetical fraud detection case with two different probability functions defined on the time intervals to study its impact.

4 Determining Correlated Events in a Fraud Detection Case

In this section, we apply the model on a hypothetical fraud detection case to demonstrate its usefulness in determining correlated events from different sets of sequences. The case involves a series of packet captures on a suspicious subnet which was detected to generate malicious traffic. In particular, one of the users within the subnet was observed to mirror a national bank website and host it subverting the firewall in an attempt to phish for personal users information from genuine bank customers. The sequence of captures was determined to contain sets of ARP, DNS, UDP, TCP, HTTP and IRC traffic. The network structure based on forensic analysis is presented in figure 1.

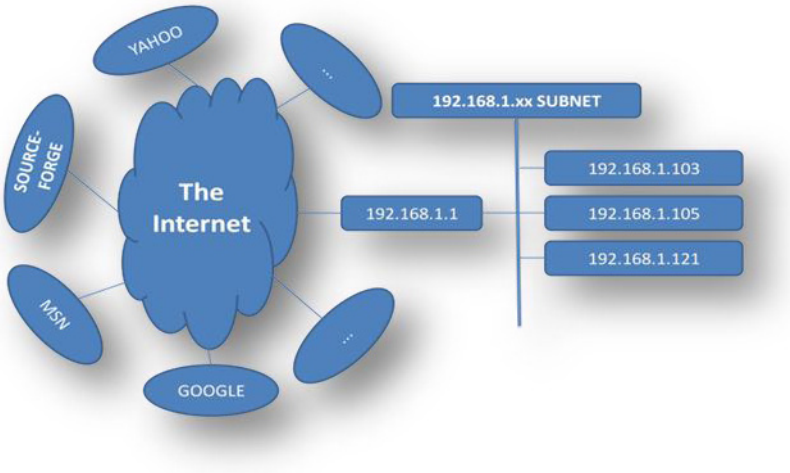

Fig. 1. Network structure based on reconstruction

Based on prior case information, we set the values of $\Delta = 5$ min and $\delta = 2$ min. The collected packets were organized into three classes of traffic, namely, DNS, UDP/TCP/HTTP and IRC sessions. Each sequence was intrinsically time ordered and synchronized with the same clock. We have experimented the correlation function with two separate probability functions P_1 and P_2.

$$P_1\ (t_j - t_i) = \delta\ /\ [\delta + \alpha(t_j - t_i)]$$
$$P_2\ (t_j - t_i) = e^{-|tj - ti|}$$

We define the events of interest in the sequence of activities as follows:

e_1: DNS request for www.google.com
e_2: HTTP request on Google search "how to mirror website?"
e_3: HTTP request on Google search "wget win32 binary"
e_4: Transfer session established with users.ugent.be/~bpuype/wget/
e_5: First TCP packet in sequence between client and wireless access point

e_6: Execution of firewall.sh and cgi-bin/webif.sh scripts on the wireless access point

e_7: IRC session between the suspicious client and Yahoo messenger server

e_8: First TCP packet tunneled on unknown port number

The respective time instants of occurrence are captured in the table 1. In our experiments, we arbitrarily set the first event from DNS traffic as the DNS request for Google server. While there were several hits on the HTTP traffic, the most important packet of relevance to the case was determined to be a Google search query from the suspicious client for "how to mirror websites?" which was assigned e_2. Then this packet was maintained as a reference point and we mined for a correlation on the other source. This proved to be another DNS request for Google server which was reverse correlated to the HTTP traffic to a HTTP packet requesting Google search for "wget win32 binary" which was assigned e_3. By repeating this procedure, we determined that the next interesting event was a DNS query to users.ugent.be/~bpuype/wget/ followed immediately by a HTTP session with that server. Keeping the case in mind, we assigned the first HTTP packet exchanged with client as e_4.

The case actually contained some interesting TCP sessions between the client and a machine determined to be the wireless access point in the subnet. We arbitrarily assigned the first such packet as event e_5. Prior to this the correlation died out between the previous determined events and any subsequent packets on the network and was mostly along expected lines. When the correlation process was repeated, we observed one HTTP packet containing scripts later determined to reconfigure firewall executing on the access point and was assigned e_6. This was again repeated but no significant correlation was detected on the DNS sequence.

We then analyzed the IRC session and tagged the client communication with the Yahoo messenger server as event e_7. This allowed us to correlate back with the HTTP

Table 1. Table reporting the time of occurrence of event in the fraud detection case.

Events	Time instants (time of day format)
e_1	14:09:59:416910
e_2	14:10:15:155434
e_3	14:10:35:053197
e_4	14:12:17:558751
e_5	14:29:04:457252
e_6	14:29:04:602225
e_7	14:30:42:102514
e_8	14:30:55:457066

sequence which enables to determine the first packet being tunneled on an unknown port. This was tagged as event e_8.

5 Implication of the Model to Correlation

The definition of $P_1(t_j - t_i)$ was motivated by the thought that the probability should be linear in Δt and inversely related to the size of the interval. The constant α is a scaling factor which was set to 1 while computing the probabilities using P_1. When the probabilities were computed using $P_2(t_j - t_i)$, we normalized the value of the time interval with δ to obtain numerically significant values when can then be compared. The table reporting the calculated probabilities between the pairs of correlated events is given in table 2. While the use of Δ was not directly evident in the calculation of the probabilities, it sets a window of observation time within the traffic packets that enabled the determination of the events e_2, e_3, e_4, e_6 and e_8.

While this paper focuses on determining correlations across different sources, in this particular example that amounted to merely determining the arrival of the next DNS request to a new web server. As the DNS requests themselves cannot imply criminal activity or malicious behavior, we have modified the definition of correlation in this context to determine occurrence of packets with relevance to this case. It so happens that they are subsequent HTTP packets, already arranged in a sequence. This brings us to an interesting juncture. This ability to dynamically modify the definition of correlation from multiple sources to within the same source makes this technique rather powerful in dealing with singlet sequences which are very large in size. If one can define specific points of interest as we have defined in this case, it allows an investigator to then focus one's attention around such events and determine correlated events which occur within the window defined by Δ and δ. Having determined these correlated events, it is then worthwhile to refine these windows of observation and drill down to the details of which particular packets are incriminating in nature and in what form.

Table 2. Table reporting the correlation probabilities for pairs of events in the fraud detection case

Corr. events	Time interval (s)	$P_1(t_j\text{-}t_i)$	$P_2(t_j\text{–}t_i)$
$C(e_1, e_2)$	15.738524	0.884053	0.877082
$C(e_2, e_3)$	19.897763	0.857769	0.847203
$C(e_3, e_4)$	102.505554	0.539312	0.425618
$C(e_5, e_6)$	0.144973	0.998793	0.998793
$C(e_7, e_8)$	13.354552	0.899857	0.894681

6 Conclusions and Future Work

In this paper, we presented an evidence composition model based on the time of occurrence of events. The time of event is a mathematically comparable quantity which is then used to compute time intervals between pairs of events (e_i, e_j). A correlation function is defined over the time interval and predefined thresholds allow us to determine the probability that a pair of events is correlated. The probability function can vary from a simple linear function to complex non-linear functions. The concept was applied to a fraud detection case with two different probability functions defined over the time interval to demonstrate its applicability. The apparent synonymy between correlation and probability functions will clear up and pave the way for clarity in their usage when we apply this model to large number of cases and learn from them.

In future, we propose to apply this evidence composition model to a more comprehensive list of evidence sources. This paper only explores the concept of correlation function using some basic probability functions. In future we expect to validate more complex functions which determine correlations over larger time ranges. Tackling the challenge of quick searches across such sets of sources is an equally challenging task. We believe that better heuristics and domain knowledge would provide more efficient solutions.

References

1. Bogen, A.C., Dampier, D.A.: Unifying Computer Forensics Modeling Approaches: Engineering Perspective. In: Proceedings of the First Intl. Workshop on Systematic Approaches to Digital Forensic Engineering (SADFE 2005). IEEE Publication, Los Alamitos (2005)
2. Carrier, B.D., Spafford, E.H.: Categories of digital investigation analysis techniques based on the computer history model. Digital Investigation. In: The Proceedings of the 6th Annual Digital Forensic Research Workshop (DFRWS 2006), vol. 3(supplement 1), pp. 121–130 (2006)
3. Case, A., Cristina, A., Marziale, L., Richard, G.G., Roussev, V.: FACE: Automated digital evidence discovery and correlation, Digital Investigation. In: The Proceedings of the Eighth Annual DFRWS Conference, September 2008, vol. 5(Supplement 1), pp. S65–S75 (2008)
4. Cohen, M.I.: PyFlag - An advanced network forensic framework, Digital Investigation. In: The Proceedings of the Eighth Annual DFRWS Conference, September 2008, vol. 5(Supplement 1), pp. S112–S120 (2008)
5. Gladyshev, P., Patel, A.: Finite state machine approach to digital event reconstruction. Digital Investigation 1(2), 130–149 (2004)
6. Hosmer, C.: Digital evidence bag. Communications of the ACM 49(2), 69–70 (2006)
7. Jeyaraman, S., Atallah, M.J.: An Empirical Study of Automatic Event Reconstruction Systems, Digital Investigations. In: Proceedings of the 6th Annual Digital Forensic Research Workshop (DRFWS 2006), vol. 3(Supplement 1), pp. S108–S115 (2006)
8. Raghavan, S., Clark, A.J., Mohay, G.: FIA: An Open Forensic Integration Architecture for Composing Digital Evidence. In: Forensics in Telecommunications, Information and Multimedia. LNCS Series on Social Informatics and Telecommunications Engg., vol. 8, pp. 83–94. Springer, Heidelberg (2009)
9. Wang, W., Daniels, T.E.: Network Forensic Analysis with Evidence Graphs. Paper presented at the 5th Annual Digital Forensic Research Workshop, DFRWS 2005 (2005)