Optimizing Live Digital Evidence Mining Using Structural Subroutines of Apriori Algorithm

Akshay Zadgaonkar
Department of Computer Science & Engineering
G.H. Raisoni College of Engineering
Nagpur, India

Ms. Vijaya Balpande
Department of Computer Science & Engineering
G.H. Raisoni College of Engineering
Nagpur, India

Abstract— The Scope and Complexity of the Internet has grown exponentially. This growth has made digital forensic investigation a very challenging task. Even the modest intra-organizational networks have sufficient network traffic to pose a problem for digital crime investigators to police and collect evidences. Another problem in Network based Crime Investigation is that Offline Mining Techniques do not yield pervasive evidence. At the same time due to voluminous traffic, live evidence mining becomes a challenge. This paper presents a technique to optimize the live evidence mining by using the principles of apriori algorithm to trigger the evidence collection mechanism at right and opportune moment. The crux of this technique is answering "When & What Information" to Collect about a subject of investigation or Data.

Keywords-Evidence Mining;Live Forensics;Aprori;

I. Introduction

Digital Forensics can formally be defined as the technique of preservation, identification, extraction, documentation and interpretation of Network and Computer Based Evidences that help in answering following questions [1]:

a. Determine What Happened

b. The Extent of the Problem

c. Determine who was responsible

In any forensic Investigation, the most critical part is evidence collection. Strength of any investigation depends on the quantity and credibility of the evidence collected. The problem in investigation of Network based crime is that the Data Space is very large. The challenge here is to determine and identify the data or piece of information (from trillions of bytes of network traffic) that could prove to be credible evidence as the golden rule of any investigation is that it should treated as if it will end in court.

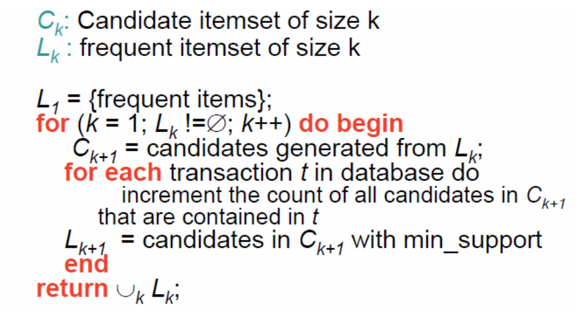
Offline Investigation suffers from various challenges like Big and Higher Capacity disks, Difficulty in legal seizure of systems and the fact that some data or evidences reside only in RAM or as Network Traffic. Due to all these reasons live evidence collection (evidence collection done pro-actively when the system is not shut and is up and running) is adopted. Due to extremely high traffic and data volume, there is a need to optimize this evidence collection. It is not possible to collect and store each and every data assuming that it might be evidence later. The evidence collection has to be triggered at right time. The technique presented in this paper identifies occurrence of an illegal activity, if threat probability is established, auto-evidence collection is triggered.

Martim d'Orey et. al. proposed a technique that used Honeypots[2] along with interceptor modules for capturing live evidences over a network. It was found that when honeypot was scaled up, Interceptor module degraded the performance and created noticeable overhead due to large traffic. What the technique lacked was intelligent interception. Antonio Savoldi et. al. highlighted another important problem associated with live evidence mining; The problem of blurring of live evidences due to the footprints of intercepting toolkit[3].

This paper presents a solution to these problems by optimization of evidence mining using Selective Interception using the subroutines Confidence() and Support() employed by Apriori Algorithm. Developed by Agrawal and R. Srikant[4], Apriori is an effective algorithm that solves the problem of discovering frequent item sets in a large database and mining association rules from frequent item sets. The key to optimizing any evidence collection mechanism is to reduce redundancy. So if the evidence collection and analysis is done only on relevant data, the probability of securing credible and pervasive evidence becomes higher**.**

II. THE APRIORI ALGORITHM

Apriori generates frequent itemsets based on two auxiliary functions called as Support() and Confidence(). These are the two principal elements that we would be using for optimizing the live evidence mining. The Support() function gives the probability of occurrence of a particular event and confidence() gives probability of two events occurring at same time. This algorithm is generally used in solving shopping preference problem like “What is the probability of shopper buying potato chips after he has bought potato chips”.



```
$C_k$: Candidate itemset of size k
$L_k$ : frequent itemset of size k

$L_1$ = {frequent items};
for ($k$ = 1; $L_k$ !=$\varnothing$; $k$++) do begin
    $C_{k+1}$ = candidates generated from $L_k$;
  for each transaction $t$ in database do
        increment the count of all candidates in $C_{k+1}$
      that are contained in $t$
  $L_{k+1}$ = candidates in $C_{k+1}$ with min_support
  end
return $\cup_k L_k$;
```

Subroutines used in Apriori:

Support(A) = Number of Transactions containing A) / (Total Transactions)

Confidence(A, B) = (Number of Transactions Containing Item A and Item B) / (Support(A))

Figure 1: Apriori Algorithm Pseudocode

As an Example, Consider the Transactions given below:

TABLE I. SAMPLE DATA

Sr. No.	***Items***
1	Shoes, Shirt, Jacket,
2	Shoes, Jacket
3	Shoes, Jeans
4	Shirt, Sweatshirt

Support(A) = Number of Transactions containing A) / (Total Transactions)

(***Support****(Shoes)* = *75%*)

Confidence(A, B) = (Number of Transactions Containing Item A and Item B) / (Number of Transactions containing A)

*(***Confidence(**Shoes, Jacket) = 66%)

Table 1 stores set of frequently purchased items. If we study the *Support()* and *Confidence()* subroutines we find that *Support()* gives us the measure of the basic occurrence of an event. The *confidence()* subroutine gives the measure probability of occurrence of event B if event A has already occurred.

The ability of Support and Confidence function to elaborately quantify the event occurrence is used in this research of optimizing evidence Collection.

III. The Evidence Collection Trigger framework: Model and Technique

The proposed technique uses a structured knowledge-base as source of information about activities that are considered contraband or illegal. Construction of the knowledge base is very easy. In case of intra-organizational network, the database of banned activities and keywords can be configured by the system administrator. Sequence of actions that might lead to a suspicious activity which warrants evidence collection is represented in the following way:

*{action1, action2, …., **consequence}***

We include all the actions and consequences that they might cause in one set.

TABLE II. Threat Transactions

Sr. No.	Items
1	{google, facebook, rapidshare, p2p, **torrent, download**}
2	{rapidshare, p2p, **download}**
3	{bank, Nigeria, prince, **fraud}**

Here we find a sample snippet of the threat model that is used in triggering the evidence collection. By using the auxiliary functions of support() and confidence(),the threats and occurrence of a banned consequence (in bold) are quantified

Hence if p2p and rapidshare occur, by support() and confidence() it is established that there is a high chance of an illegal download happening. Each threat is assigned a threat index. This threat index acts as a threshold for triggering evidence collection.

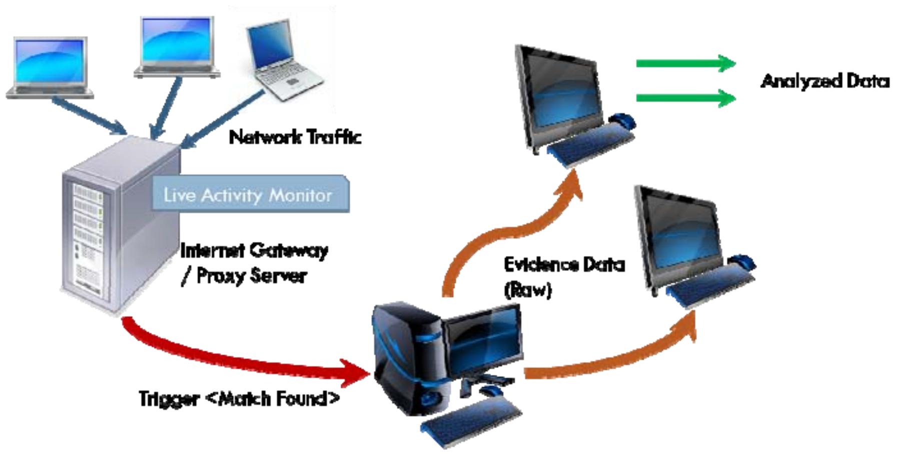
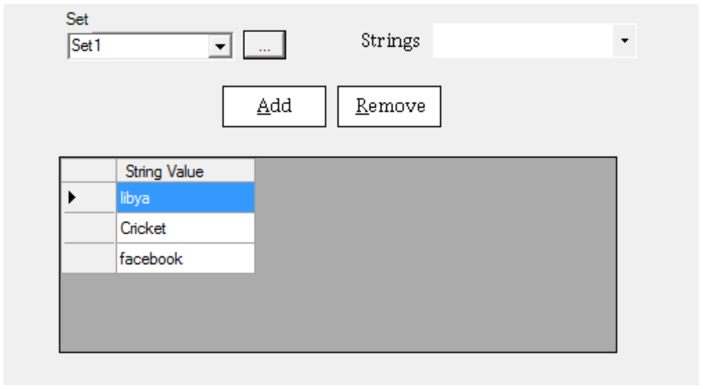


Figure 2: Evidence Collection Framework

From the figure it can be seen that the area of concern is to decide when to trigger the evidence collection and analysis module and that is where the proposed optimization technique becomes an effective way of streamlining the evidence collection process.

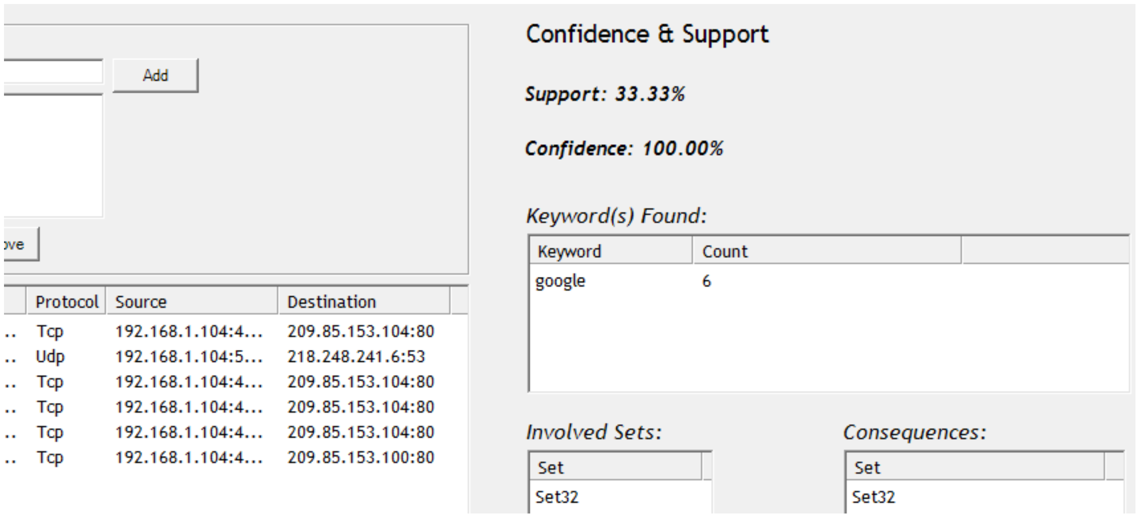
We have implemented this idea in a form of a network packet sniffer that implements this optimization technique. This Program has been designed in C#.NET using Windows APIs for communication with Networking Subsystem. It uses MySQL database for storing the *action consequence* sets and threat indexes. This application works as follows:

1) Initially the threat sets are defined. They can be either IP Based or They can be string based i.e. the sniffing would monitor based on the preference chosen
2) Define the Constituent Members of the Set. The Set is represented in the form of a action consequence list that is mentioned above.
3) Provide the Consequnce Keyword with which the Threat index has tobe matched for analysis
4) Click Start Monitering and Wait for the program to sniff the packets that contain the relevant data.
5) The support and confidence are calculated by the program and it also provides the possible consequences of sequence of actions



	String Value
▶	libya
	Cricket
	facebook

Figure 3: Defining Sets



Support: 33.33%

Confidence: 100.00%

	Protocol	Source	Destination
..	Tcp	192.168.1.104:4...	209.85.153.104:80
..	Udp	192.168.1.104:5...	218.248.241.6:53
..	Tcp	192.168.1.104:4...	209.85.153.104:80
..	Tcp	192.168.1.104:4...	209.85.153.104:80
..	Tcp	192.168.1.104:4...	209.85.153.104:80
..	Tcp	192.168.1.104:4...	209.85.153.100:80

Keyword(s) Found:

Keyword	Count
google	6

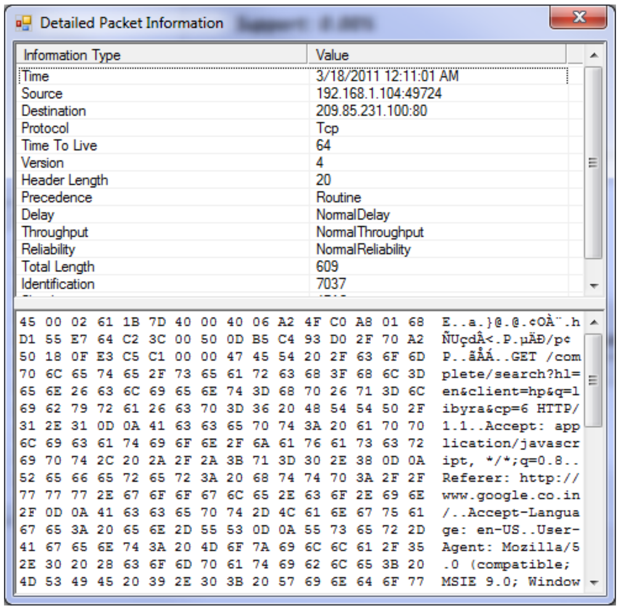
Involved Sets:

Set
Set32

Consequences:

Set
Set32

Figure 4: Main Screen



Information Type	Value
Time	3/18/2011 12:11:01 AM
Source	192.168.1.104:49724
Destination	209.85.231.100:80
Protocol	Tcp
Time To Live	64
Version	4
Header Length	20
Precedence	Routine
Delay	NormalDelay
Throughput	NormalThroughput
Reliability	NormalReliability
Total Length	609
Identification	7037

```
45 00 02 61 1B 7D 40 00 40 06 A2 4F C0 A8 01 68  E..a.}@.@.¢OÀ¨.h
D1 55 E7 64 C2 3C 00 50 0D B5 C4 93 D0 2F 70 A2  ÑUçdÂ<.P.µÄÐ/p¢
50 18 0F E3 C5 C1 00 00 47 45 54 20 2F 63 6F 6D  P..ãÅÁ..GET /com
70 6C 65 74 65 2F 73 65 61 72 63 68 3F 68 6C 3D  plete/search?hl=
65 6E 26 63 6C 69 65 6E 74 3D 68 70 26 71 3D 6C  en&client=hp&q=l
69 62 79 72 61 26 63 70 3D 36 20 48 54 54 50 2F  ibyra&cp=6 HTTP/
31 2E 31 0D 0A 41 63 63 65 70 74 3A 20 61 70 70  1.1..Accept: app
6C 69 63 61 74 69 6F 6E 2F 6A 61 76 61 73 63 72  lication/javascr
69 70 74 2C 20 2A 2F 2A 3B 71 3D 30 2E 38 0D 0A  ipt, */*;q=0.8..
52 65 66 65 72 65 72 3A 20 68 74 74 70 3A 2F 2F  Referer: http://
77 77 77 2E 67 6F 6F 67 6C 65 2E 63 6F 2E 69 6E  www.google.co.in
2F 0D 0A 41 63 63 65 70 74 2D 4C 61 6E 67 75 61  /..Accept-Langua
67 65 3A 20 65 6E 2D 55 53 0D 0A 55 73 65 72 2D  ge: en-US..User-
41 67 65 6E 74 3A 20 4D 6F 7A 69 6C 6C 61 2F 35  Agent: Mozilla/5
2E 30 20 28 63 6F 6D 70 61 74 69 62 6C 65 3B 20  .0 (compatible;
4D 53 49 45 20 39 2E 30 3B 20 57 69 6E 64 6F 77  MSIE 9.0; Window
```

Figure 5: Sniffed Packet Information

IV. AFTER THE TRIGGER: PROCESS OF EVIDENCE ANALYSIS

Once the live monitoring program generates trigger, all vital information regarding the subject, object and context is extracted and an investigative reasoning model called as Case Oriented Evidence Mining is employed for Evidence Analysis. Case Oriented evidence mining can play a vital role in narrowing the state space for searching the goal i.e. evidences. Jhun Zang et. Al. Proposed a new digital forensic process model[5], named COEM (case-oriented evidence mining model), along with its approaches, top-down. The model consists of sets of tasks described at two levels of abstraction: case (from requirement to finding) and data (from general to specific). COEM facilitates analysis in a viewpoint of case instead of data and gives investigators a more natural and rational perspective.

Evidence Analysis is done in following steps[6]:

Pre-processing

The Live Forensic toolkit needs certain intelligence and data input for smart and discrete evidence collection. For any investigation, conciseness of the evidence plays vital role in efficient investigation as search space is extremely large. In this stage we define the problem. Based on further heuristics, the data is collected and segregated. The data is later encoded to the desired form

Modeling

Based on the data available from the forensic toolkit, the next crucial decision is the methodology to be adopted for live evidence mining i.e. whether they would be association rules or any other algorithmic approach.

Assessment

The chosen model is analyzed with sample data and trained. Its affectivity is analyzed and based on results it is implemented in the COEM framework or the activity is traced back to modeling if results are unsatisfactory.

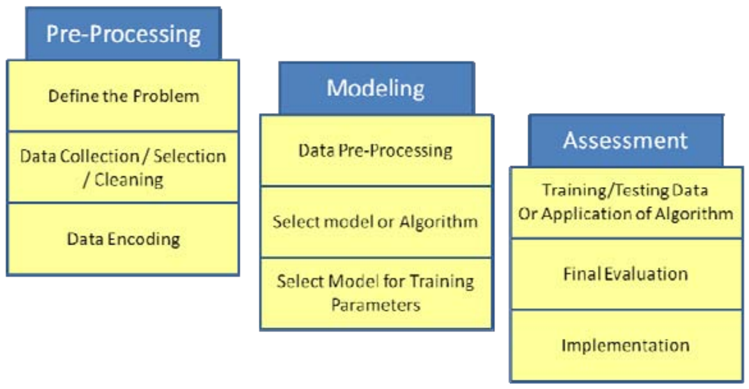

Pre-Processing
Define the Problem
Data Collection / Selection / Cleaning
Data Encoding
Modeling
Data Pre-Processing
Select model or Algorithm
Select Model for Training Parameters
Assessment
Training/Testing Data Or Application of Algorithm
Final Evaluation
Implementation

Figure 6 Evidence Mining Framework

After finalizing the above data mining framework, we implement it in the generic model and obtain data.

The next and most crucial objective is to transform this data into form that is admissible by the court of law. The data is fashioned in the following form and each investigational document is transformed to match the following template.

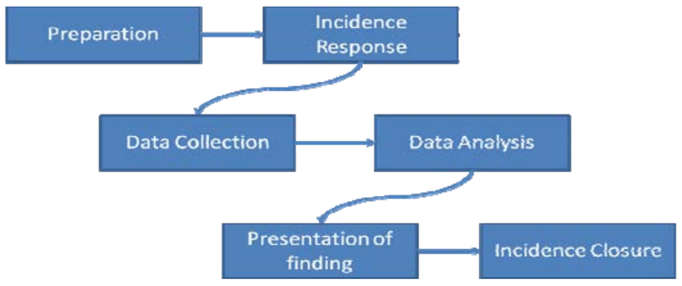


Figure 7: Chronicle of Investigation Documentation

After preparation and transformation of the raw data into systematic form, all the gathered evidences are analyzed for their credibility. If unsatisfactory, the investigation processes might backtrack to previous steps. This primary Collection procedure can be appended with supplementary evidence collection and analysis frameworks like Highly Extensible Network Packet Analysis (HENPA) framework[7], which takes the output of a packet sniffer and processes the data to extract potential forensic evidence.

V. CONCLUCSION & CHALLENGES AHEAD

The confidence and support functions optimize the process of evidence collection. It is paramount that Investigation collects evidences that are relevant, pervasive and succinct. The proposed technique does that by letting the evidence analysis system work on *only* the data that is relevant. This selective analysis accelerates the auto-evidence generation and makes it technically and logistically feasible.

The major challenge here is improvement in trigger propagation and false positives. The future work concentrates on improving the accuracy and transforming this lab scale program to a complete, scalable and robust evidence mining tool.

References

[1] Michael Gleeson, David Markey, Dr. Fred Mtenzi- Investigation and Development of a Security and Forensic Analysis - IEEE Journal Publication - pg. 373 – 378

[2] Martim d'Orey Passer de Andrade Carbone, Paulo Licio de Ceus - A Mechanism for Automatic Digital Evidence Collection on High-Interaction Honeypots - Proceedings of the 2004 IEEE Workshop on Information Assurance and Security United States Military Academy, West Pohlt, NY, 10-11 June 2004

[3] Antonio Savoldi & Paolo Gubian, Isao Echizen - How to Deal with Blurriness in Live Forensics: A Case of Study - 2009 Fifth International Joint Conference on INC, IMS and IDC - Pg. 1865-1871

[4] R.Agrawal and R.Srikant.Fast algolithms for mining association rules in large databases.In Research Report RJ 9839,IBM Almaden Research Center,San Jose,CA,June 1994

[5] Jun Zhang, Lina Wang, Application of Case-oriented Evidence Mining in Forensic Computing, 2009 International Conference on Multimedia Information Networking and Security. pg 103-106

[6] Jae Hoon Sun1, Hyun Seok Yoon1, Jae Hyung Yoo1 - Design and Implementation of KFMS for Digital Forensics - 2008 International Conference on Information Science and Security - pg.214-219

[7] Joshua Broadway, Dr Benjamin Turnbull, and Associate Professor Jill Slay, Member, IEEE Improving the Analysis of Lawfully Intercepted Network Packet Data Captured For Forensic Analysis- The Third International Conference on Availability, Reliability and Security -pg. 1361-1368

AUTHORS PROFILE

Akshay Zadgoankar finished his Bachelor of Engineering (B.E.) degree from G.H. Raisoni College of Engineering (An Autonomous Instiutue, Accreditated by NBA). He is currently pursuing M.Tech (Masters) Course in Computer Science.

Ms. Vijaya Balpande finshed her M.Tech. Course in Computer Science from Nagpur University. She is currently pursuing her Ph.D. She is also an authorized guide for students pursuing Masters Degree in Computer Science