International Journal of Information & Network Security (IJINS)
Vol.3, No.5, October 2014, pp. 324~335
ISSN: 2089-3299

A Data Mining Approach for Predicting Attacks and Recognizing Threat Strategy in the context of Collaborative Attackers and Network Security Management

Oluwafemi Oriola*, Adesesan Barnabas Adeyemo, Oluwaseyitanfumi Osunade***
* Departement of Computer Science, Adekunle Ajasin University, Akungba Akoko, Nigeria
** Department of Computer Science, University of Ibadan, Ibadan, Nigeria

Article Info

Article history:

Received Jun 12th, 2014
Revised Aug 20th, 2014
Accepted Aug 26th, 2014

Keyword:

Internet-facilitated Threat,
Collaborative Network Security Management ,
Sequential Association Mining,
Attack Prediction,
Threat Prediction Model

ABSTRACT

Data Mining has been novel in predictive modeling. In fact, various Data Mining Models have been used to predict future attacks and recognize threat strategy. However, none has been applied to predict attack and recognize threat strategy in the context of Collaborative Attacker and Victim. In recent times, Internet-facilitated Threats such as botnet and advanced persistent threats have been responsible for most successful attacks in organisations while multiple targets have been the victims of the attacks. Hence, this paper presents a Data Mining Approach for predicting attacks and recognizing threat strategy in the context of Collaborative Attacker and Victim Systems. An Actionable Sequential Association Data Mining Model is developed to mine attack sequences from a repository of Central Administrative System. Plymouth University and MIT Lincoln Lab LLDOS 1.0 Attacker and Victim scenarios are used to evaluate the model. The predictability of the Data Mining Model records 100% accuracy in all scenarios examined. This shows that threats in the context of Collaborative Attackers and Victims are better predicted using Threat Prediction Model that incorporatesactionable attributes and context into data mining.

Corresponding Author:

Oluwafemi Oriola,
Departement of Computer Science,
Adekunle Ajasin University,
Akungba Akoko, Ondo State, Nigeria.
Email: oluwafemioriola@yahoo.com

1. INTRODUCTION

Internet is one of the greatest innovations that has benefitted human race since nineteenth century. It has closed the boundary among the various aspects of societies; now, it can be accessed everywhere via web, mobile, or cloud. Apart from the individual usage of internet, groups of users do benefit from it. Particularly, some Attackers collaborate via internet to exploit the vulnerability of victim systems and cause damage in organized manner. This kind of threat is referred to as Internet-facilitated Organized Threat and they make use of three common methods in achieving their missions: Botnet, Worm Propagation and Advanced Persistent Threat (APT).

In a typical organization operating over internet, the Demilitarized Zone (DMZ) is protected by limited security configurations while the inside zone is protected with more effective security configurations (Paquet, 2013). A successful compromise of security in DMZ could lead to security compromise in the inside zone. Also, the

vulnerability of one asset could lead to the vulnerability of other assets in the same zone or different zones depending on the configuration. This means that attack in organizations could be better described as multistage phenomenon (Gomez, 2011) instead of single stage phenomenon. This means that a Network Security domain is violated by multiple stages of attack referred to as Scenario Attack. In Collaborative Victim System like Collaborative Network Security Management, different security domains are identified, assessed and monitored. Hence, different sensors' events are analyzed either centrally or in the distributed domains with a single control. Attack Prediction and Threat strategy recognition is one of the activities carried out by a Collaborative Network Security Manager. Attack Prediction is the forecasting of next successful point of action or attack that will be perpetrated by attacker. Threat Strategy Recognition deals with the recognition of plausible path(s) attacker would follow in compromising the security state of victim system among all the likely attack paths. Collaborative Network Security Management is faced with the task of predicting Complex Scenario Attack. This work therefore examines the ability of Data Mining in predicting scenario of attack and recognizing threat strategy in Collaborative Network Security Management.

Despite the predictive ability of data mining, only few works have applied Data Mining in predicting attack and recognizing threat strategy. Alert correlation method that targets the automated construction of attack graphs from a large volume of raw alerts was developed based on multi-layer perceptron (MLP) and Support Vector Machine (SVM) Zhu and Ghorbani (2006). They used the probability output of MLP or SVM to connect correlated alerts in a way that they represent the corresponding attack scenarios. Data mining approach was applied in generating attack graphs in Li *et al.* (2007). Through Association Rule Mining, the algorithm generated multi-step attack patterns from historical intrusion alerts which comprised the attack graphs. The algorithm also calculated the predictability of each attack scenario in the attack graph which represented the probability for the corresponding attack scenario to be the precursor of future attacks.

Other approaches employed in predicting threats are reviewed below. The following review discusses the approaches. Wang et al. (2006) relied on finite memory, where the index can only be built on a limited number of alerts inside a sliding window. They developed a novel queue graph (QG) approach, in which instead of searching all the received alerts for those that prepare for a new alert, they only searched for the latest alert of each type. The correlation between the new alert and other alerts was implicitly represented using the temporal order between alerts. The approach could correlate alerts that are arbitrarily far away in linearly efficient time. Then, they extended the basic QG approach to a unified method to hypothesize missing alerts and to predict future alerts. Nanda and Deo (2007) presented a technique to identify attacks on large networks using a highly scalable model, while filtering for false positives and negatives. It also forecasts the propagation of the security failures proliferated by attacks over time and their likely targets in the future.

Pandey *et al.* (2008) identified the potential algebraic properties of capability in terms of operations, relations and inferences. The properties gave better insight to understand the logical association between capabilities which will be helpful in making the system modular. They also presented variant of correlation algorithm by using the algebraic properties. Jemili *et al.* (2009) proposed an Intrusion Detection and Prediction System based on uncertain and imprecise inference networks and its implementation. In our contribution, they chose to do a supervised learning based on Bayesian networks. The choice of modelling the historic of data with Bayesian networks is dictated by the nature of learning data (statistical data) and the modeling power of Bayesian networks. However, taking into account the incompleteness that can affect the knowledge of parameters characterizing the statistical data and the set of relations between phenomena, the proposed system in the present work uses for the inference process a propagation method based on a bayesian possibilistic hybridization.

Li and Tia (2010) developed the intrusion alerts correlation. The multi-agent system architecture consisted agents and sensors, the sensors collected security relevant information, and the agents processed the information. The State Sensor collected information about security state and the Local State Agent and Centre State Agent pre-processed the security state information and converted it to ontology. The Attack Correlator correlated the attacks and outputs the attack sessions. Haslum (2010) developed a Probabilistic Hidden Markov Model (HMM) that captures the interaction between the attacker and the network. The interaction between various Distributed IDS and integration of their output were achieved through a HMM. They modelled the interaction between the attackers and the system using a Markov model and assumed the system to be in one of the following states: Normal (N) indicating that there is no on-going suspicious activity, Intrusion Attempt (IA) indicating suspicious activity against the network, Intrusion in Progress (IP) indicating that one or more attacker have started an attack against the system, and Successful Attack (SA) one or more attackers have already broken into the system. By using a Markov model, they assumed that next state transition only depend on current state.

Another data mining technique to discover, visualize, and predict behavioural pattern of attackers in a network based system was developed by Katipally et al. (2010). They proposed a system that was able to discover temporal pattern of intrusion which revealed behaviours of attackers using alerts generated by Intrusion Detection System (IDS). They used data mining techniques to find the patterns of generated alerts by generating Association rules. Their system was able to stream real-time Snort alerts and predict intrusions based on our learned rules. Farhadi et al. (2011) combined Data Mining and HMM to predict threat. In this paper, they proposed an alert correlation system consisting of two major Components: an Attack Scenario Extraction Algorithm (ASEA), which mines the stream of alerts for attack scenarios. The ASEA had a relatively good performance, both in speed and memory consumption. Contrary to previous approaches, the ASEA combines bothprior knowledge as well as statistical relationships and Hidden Markov Model (HMM)-based correlation method of intrusion alerts, from different IDS sensors across an enterprise. They used HMM to predict the next attack class of the intruder, also known as plan recognition.

2. METHODS

2.1 Data Mining

The methodology used is adapted from Data Mining, refers to as nontrivial extraction of implicit, previously unknown and potentially useful information from data in databases (Zaiane, 1999). It is a key step of knowledge discovery in databases (KDD) which is the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data (Fayyad *et al.,* 1996). In other words, data mining involves the systematic analysis of large data sets using automated methods. The Knowledge Discovery in Databases contains the following steps as presented in Figure 1.

i. *Developing an understanding of the application domain.* This is the initial preparatory step. It prepares the scene for understanding what should be done with the many decisions (about transformation, algorithms, representation, etc.). The people who are in charge of a KDD project need to understand and define the goals of the end-user and the environment in which the knowledge discovery process will take place (including relevant prior knowledge). As the KDD process proceeds, there may be even a revision of this step. Having understood the KDD goals, the pre-processing of the data starts.

ii. *Selecting and creating a data set on which discovery will be performed*. Having defined the goals, the data that will be used for the knowledge discovery should be determined. This includes finding out what data is available, obtaining additional necessary data, and then integrating all the data for the knowledge discovery into one data set, including the attributes that will be considered for the process. This process is very important because the Data Mining learns and discovers from the available data. This is the evidence base for constructing the models. If some important attributes are missing, then the entire study may fail.

iii. *Pre-processing and cleansing.* In this stage, data reliability is enhanced. It includes data clearing, such as handling missing values and removal of noise or outliers. It may involve complex statistical methods or using a Data Mining algorithm in this context. For example, if one suspects that a certain attribute is of insufficient reliability or has many missing data, then this attribute could become the goal of a data mining supervised algorithm. A prediction model for this attribute will be developed, and then missing data can be predicted.

iv. *Data transformation.* In this stage, the generation of better data for the data mining is prepared and developed. Methods here include dimension reduction (such as feature selection and extraction and record sampling), and attribute transformation (such as discretization of numerical attributes and functional transformation). This step can be crucial for the success of the entire KDD project, and it is usually very project-specific. Having completed the above four steps, the following four steps are related to the Data Mining part, where the focus is on the algorithmic aspects employed for each project:

v. *Choosing the appropriate Data Mining task.* We are now ready to decide: which type of Data Mining to use, for example, classification, regression, or clustering? This mostly depends on the KDD goals, and also on the previous steps. There are two major goals in Data Mining: prediction and description. Prediction is often referred to as supervised Data Mining, while descriptive Data Mining includes the unsupervised and visualization aspects of Data Mining. Most Predictive Data Mining techniques are based on inductive learning, where a model is constructed explicitly or implicitly by generalizing from a sufficient number of training examples. The underlying assumption of the inductive approach is that

the trained model is applicable to future cases. The strategy also takes into account the level of meta-learning for the particular set of available data.

vi. *Choosing the Data Mining algorithm.* Having the strategy, we now decide on the tactics. This stage includes selecting the specific method to be used for searching patterns (including multiple inducers). This approach attempts to understand the conditions under which a Data Mining algorithm is most appropriate. Each algorithm has parameters and tactics of learning (such as ten-fold cross-validation or another division for training and testing).

vii. *Employing the Data Mining algorithm***.** Finally, the implementation of the Data Mining algorithm is reached. In this step we might need to employ the algorithm several times until a satisfied result is obtained, for instance by tuning the algorithm's control parameters.

viii. *Evaluation.* In this stage, we evaluate and interpret the mined patterns (rules, reliability etc.), with respect to the goals defined in the first step. Here we consider the preprocessing steps with respect to their effect on the Data Mining algorithm results (for example, adding features in Step 4, and repeating from there). This step focuses on the comprehensibility and usefulness of the induced model. In this step the discovered knowledge is also documented for further usage. The last step is the usage and overall feedback on the patterns and discovery results obtained by the Data Mining:

ix. *Using the discovered knowledge***.** The knowledge is now ready to be incorporated into another system for further action. The knowledge becomes active in the sense that we may make changes to the system and measure the effects. Actually the success of this step determines the effectiveness of the entire KDD process.

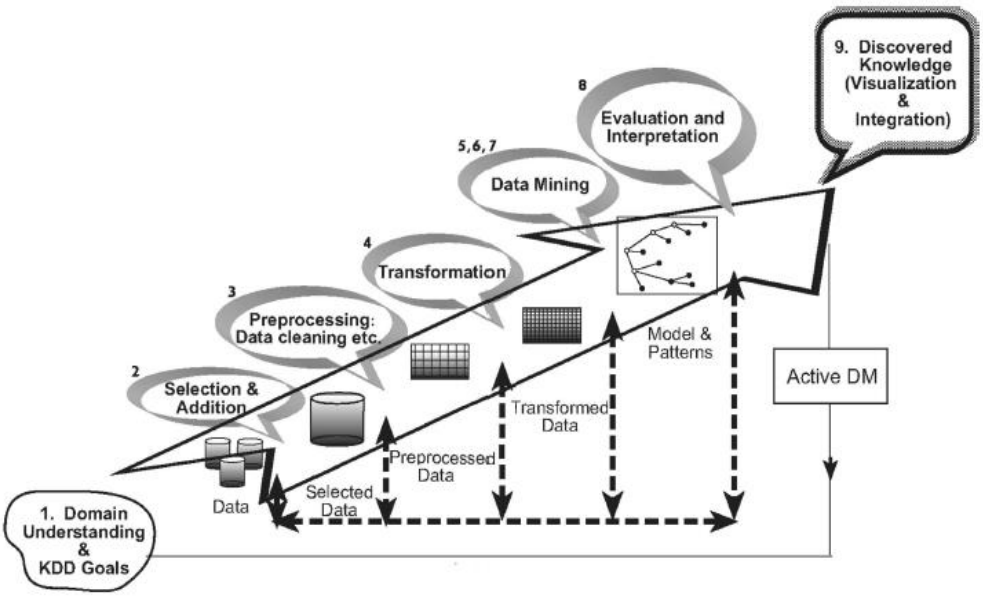

Figure 1: Processes in Knowledge Discovery in Databases for Data Mining

(Adapted from Maimon and Rokash, 2009)

2.2 Sequential Association Data Mining Model for Attack Prediction and Threat Strategy Recognition

The proposed model has three parts:

i. *Data Pre-processing*
ii. *Sequential Association Generation*
iii. *Rule Interestingness Estimation*

i. Data Pre-processing

In preprocessing, only the actionable features presented in Table 1 are selected from the alert features.

Table 1: Sample Record for Data Mining

Time	Source_IP	Destination_IP	Event
5:30	176.28.34.20	182.28.56.04	Bufferoverflow
6: 46	130.50.20.05	182.28.56.04	Bufferoverflow
7:55	176.28.34.11	182.28.75.11	Trojan

ii. Sequential Association Generation

The Temporal Association Data Mining is used to generate sequential association sequences. The illustration below describes the method used to generate the sequences.

Suppose that $x_1, x_2, \ldots, x_n$ is a stream of events. Using Sliding Window Approach similar to Li *et al.* (2007) and Farhadi *et al.* (2011), once the algorithm is run with a time-based window, the window "slides" Δ alerts in the stream ($1 \leq \Delta \leq L$). That is, if

$$[\alpha_i, \alpha_{i+1}, \ldots, \alpha_{i+L-1}]$$ is a window,

the next window will be

$$[\alpha_{i+\Delta}, \alpha_{i+\Delta+1}, \ldots, \alpha_{i+\Delta+L-1}]$$

Such that any two adjacent windows share L- Δ alerts.
In Figure 2, a Typical Window with size W_i is illustrated.

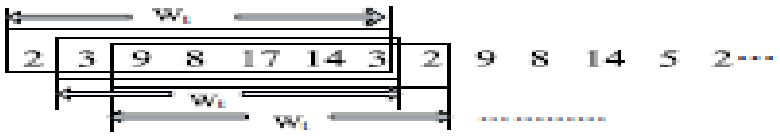

Figure 2: Illustration of a Typical Window

The following algorithm represented procedurally is used to generate the association sequences:

Step 1: Set Window size to P, SequenceSize to 1, MaximumSequence Size to L, Sequence to empty
Step 2: Sort Incidents based on their timestamps.
Step 3: Set the current WindowStep to 1.
Step 4: Set Temp to empty.
Step 5: Store Incidents according to WindowStep in Temp
Step 6: If Sequence Size is L, Continue otherwise Go to Step 9
Step 7: Increment WindowStep by 1
Step 8: Repeat Step 4 and 6
Step 9: Add incident to Temp
Step 10: Add Temp to Sequence
Step 11: Return WindowStep, Sequence

iii. Rule Interestingness Analysis

The rule Interestingness Analysis is carried out using the support and the confidence of sequence. In this case, a minimum support is set while the confidences of the sequences that meet up with the minimum support produce the interestingness of the sequences.

Given that A -> B is an association, A is known as Antecedent and B is known as Consequent. The Support and the Confidence of the Consequent given the Antecedent can be statistically calculated as presented in Equation 1 and 2. The predictability is the percentage of attacks that are correctly predicted as attack. It is based on the threats. This presented in Equation 2.

Support (B) = n(A U B) / N ... *(1)*

Confidence (B) = n (A U B) / n (A) ... *(2)*

Predictability = Confidence x 100 ... *(3)*

The following algorithm represented procedurally is used to generate the association sequence interestingness:

Step 1: *Assign MinimumSupport to MinSup, WindowStep to Max*
Step 2: *Set WindowStep to 1*
Step 3*: Set TempLocation to 0, Temp to empty*
Step 4*: While WindowStep < Max*
Step 5: Increment the WindowStep
Step 6: *Add Sequence by WindowStep to Temp*
Step 7: *If TempLocation != Temp Then Increment the TempLocation*
Step 8*: Compute the Support of Temp*
Step 9: *While Support ≥ MinSup, Compute the Confidence*
Step 10*: Assign Confidence to Interestingness*
Step 11*: Return WindowStep, Sequence, Interestingness*

2.3 EXPERIMENTAL DESIGN

Figure 3 presents the testbed for setting up the experiment. There are four network security domains. The events flowing through the domains are monitored by intrusion detection server which is managed by unified threat management system. The central administrative system collects the event and incident information from the server, carries out in-depth analysis on the server and sends the results to the network security managers in each domain.

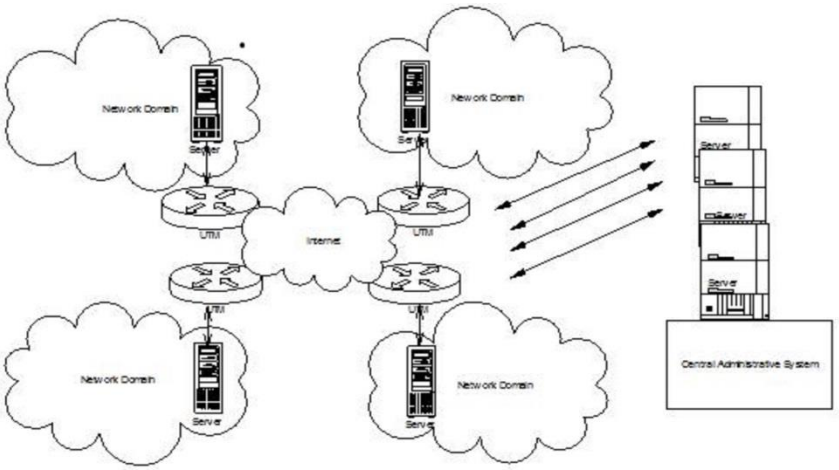

Figure 3: Collaborative Network Security Management System

In order to evaluate the model, Plymouth University Advanced Persistent Threat and MIT Lincoln Lab LLDOS 1.0 Inside Threat are used.

The Plymouth University Persistent Threat was launched to exploit Java Exploit CVE-2012-4681 in Microsoft Windows Server 2012 in four subnets 10.1.0.128/27, 10.1.0.160/27, 10.1.0.192/27 and 10.1.0.224/27.

The Attack Phases include:

i. *Connect to the Victims*
ii. *Scan the operating systems for exploitable vulnerability*
iii. *Attempt to exploit CVE-2012-4681*
iv. *Exploit CVE-2012-4681*
v. *Install Backdoors*

We replayed the LLDOS 1.0 with DARPA 1999 background traffics three times separately against Snort and Suricata Network Intrusion Detection and Prevention System. The Attack Phases include:

i. *IPsweep: Sending ICMP echo-request for live hosts*
ii. *Probe: Probe of live IP's to look for the sadmind daemon running on Solaris Hosts*
iii. *Break-in: Break-ins via the sadmind vulnerability, both successful and unsuccessful on those hosts.*

iv. Install Virus: Installation of the Trojan mstream DDoS software on three hosts using telnet.DDos:
v. Launching the DDoS attacks

3. RESULT AND ANALYSIS

Table 2 and Table 3 present the time of each replay with the size of the packets for Plymouth University and MIT Lincoln Lab data respectively. The replay lasted for average of 4minutes and 8minutes respectively.

Table 2: Plymouth University Packet Replay

Replay	Size of Packet	Date	Time	
			Snort	Suricata
Replay 1	201,307kb	21/07/2014	19:40:49-19:43:10	19:40:09-19:43:43
Replay 2	201, 307kb	21/07/2014	19:43:45-19:47:10	19:44:29-19:47:43
Replay 3	201,307kb	21/07/2014	19:47:12-19:50:10	19:48:50-19:54:43

Table 3: MIT Lincoln Lab Packet Replay

Replay	Size of Packet	Date	Time	
			Snort	Suricata
Replay 1	452,256kb	20/07/2014	14:03:17-14:10:56	13:30:42-13:44:17
Replay 2	452,256kb	20/07/2014	14:10:57-14:18:27	13:44:22-13:54:17
Replay 3	452,256kb	20/07/2014	14:18:28-14:21:13	14:30:42-14:44:17

Table 4 presents the Actionable Threat Path generated by the Plymouth University Threat Prediction Experiment. Figure 4 presents the Plymouth University attack graphs generated by the Threat Prediction Model.

Based on the same assumption that a once successful attack exploit would be exploited by an attacker in the near future than none successful one; only the attack sequence with full support (sequence that occur three times) are chosen to determine the actionable threat paths. Table 5 presents the Actionable Threat Path generated by the MIT Threat Prediction Experiment. Figure 5 present the attack graphs for MIT Lincoln LLLDOS 1.0 generated by the Threat Prediction Model. To compare our model performance, we present attack graphs results of Li *et al.* (2007) in Figure 6 and Figure 7.

In Table 4, five sequences of events of 6 steps with the support of 0.02654867 and confidence of 1 are selected after the interestingness analysis of the Plymouth University Event by the Central Administrator. Each of the sequence steps occur three times meaning that the attackers prefer to use the exploit because it always lead to success since an attacker will adhere to the strategy that will give him/her maximum benefit. This conforms to the earlier study that a novice attacker exploit easy-to-use kit (Bhattacharya and Ghosh, 2008). Figure 4 presents the actionable Threat Path derived from the sequences. In Table 5, 11 sequences of 12 steps with the support of 0.021897 and Confidence of 1 are selected after the interestingness analysis of the MIT Lincoln LLDOS 1.0 by the Central Administrator.

The comparison of the attack graph with the original attack description shows that the Threat Paths reflect to a large extent the attack steps. Different bots were applied at the reconnaissance IPsweep and scanning phases as shown in step 1 and Step 2. The Attack Graph shows that after a successful exploit of sadmind vulnerability in a host 172. 16.115.20 in a particular subnet, the attacker performs pings host 172.16.113.204 in another subnet. This conforms to the description in DARPA (2014). The comparison of the Threat Prediction result with previous Sequential Association Mining Technique by Li *et al.* (2007) output in Figure 6 and Figure 7 shows that the new approach is better than the Li's and co work. Li et al. (2007) did not show the loop but rather shows sequential attack path, which did not reflect well how a hacker works in real settings. Also, our Threat Prediction model recorded a very good performance with the predictability of 100% for all sequence while that of Li *et al.* (2007) recorded the highest of 26.6%. The combination of the steps in each scenario produces the Threat Strategy employed.

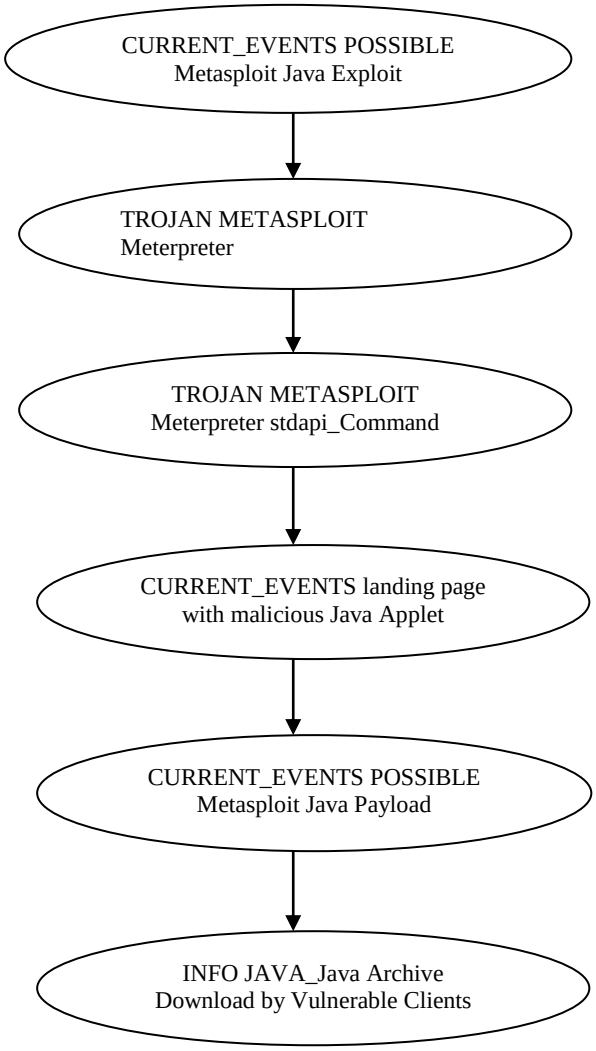

Figure 4: Plymouth University attack graph

Table 4:Actionable Threat Paths generated by the Threat Prediction Experiment for Plymouth University Attack Scenario

S/N	Attack Scenario	Exploit	Source	Destination	Frequency/ Support	Confidence
1	D2,4	CURRENT_EVENTS Possible Metasploit Java Exploit	10.1.0.3	10.1.0.135	3 times /0.02654867	1
2	D2,4=>AN2,11	Trojan Metasploit Meterpreter core_channel Command Request	10.1.0.3	10.1.0.197	3 times /0.02654867	1
3	D2,4, AN2,11 =>AO2,4	Trojan Metasploit Meterpreter stdapi_Command Request	10.1.0.3	10.1.0.135	3 times /0.02654867	1
4	D2,4, AN2,11, AO2,4 =>C2,4	CURRENT_EVENTS landing page with malicious Java Applet	10.1.0.3	10.1.0.135	3 times /0.02654867	1
5	D2,4, AN2,11, AO2,4, C2,4=>E2,4	CURRENT_EVENTS Possible Metasploit Java Payload	10.1.0.3	10.1.0.135	3 times /0.02654867	1
6	D2,4, AN2,11, AO2,4, C2,4, E2,4=>K2,4	INFO JAVA-Java Archive Download by Vulnerable Client	10.1.0.3	10.1.0.135	3 times /0.02654867	1

Table5: Actionable Threat Paths generated by the Threat Prediction Experiment for MIT Lincoln LLDOS 1.0

S/N	Attack Scenario	Exploit	Source	Destination	Frequency/ Support	Confidence
1	C12,41	INFO PING NIX	172.16.113.50	172.16.113.105	3 times /0.021897	1
2	C12, 41 =>D12,41	INFO PING BSDtype	172.16.113.50	172.16.113.105	3 times /0.0218979	1
3	C12,41, D12,41 => C10,70	INFO PING NIX	172.16.112.50	172.16.114.169	3 times /0.021897	1
4	C12,41, D12,41 C10,70 => D10,70	INFO PING BSDtype	172.16.112.50	172.16.114.169	3 times /0.021897	1
5	C12,41, D12,41 C10,70, D10,70 => M21,65	POLICY PE EXE/DLL Windows File Download	132.60.168.152	172.16.112.207	3 times /0.021897	1
6	C12,41, D12,41 C10,70, D10,70, M21,65 => A13,14	Exploit MS_SQL DOS ATTEMPT(08)	172.16.115.20	172.16.112.20	3 times /0.021897	1
7	C12,41, D12,41 C10,70, D10,70, M21,65, A13,14 => F25,31	NETBIOS NT NULL Session	172.16.116.20	172.16.112.100	3 times /0.021897	1
8	C12,41, D12,41 C10,70, D10,70, M21,65, A13,14 => F9,14	NETBIOS NT NULL Session	172.16.112.100	172.16.112.100	3 times /0.021897	1
9	C12,41, D12,41 C10,70, D10,70, M21,65, A13,14, F9,14 => K13,35	SNMP Public Access UDP	172.16.113.20	172.16.112.105	3 times /0.021897	1
10	C12,41, D12,41 C10,70, D10,70, M21,65, A13,14, F9,14, K13,35 => I20,62	RPC PORTMAP SADMIND REQUEST UDP	202.77.162.213	172.16.115.20	3 times /0.021897	1
11	C12,41, D12,41 C10,70, D10,70, M21,65, A13,14, F9,14 K13,35, I20,62 => J20,62	RPC Sadmind query with root credentials	202.77.162.213	172.16.115.20	3 times /0.021897	1
12	C12,41, D12,41 C10,70, D10,70, M21,65, A13,14, F9,14 K13,35, I20,62, J20,62 => C13,60	ICMP PING NIX	172.16.115.20	172.16.113.204	3 times /0.021897	1

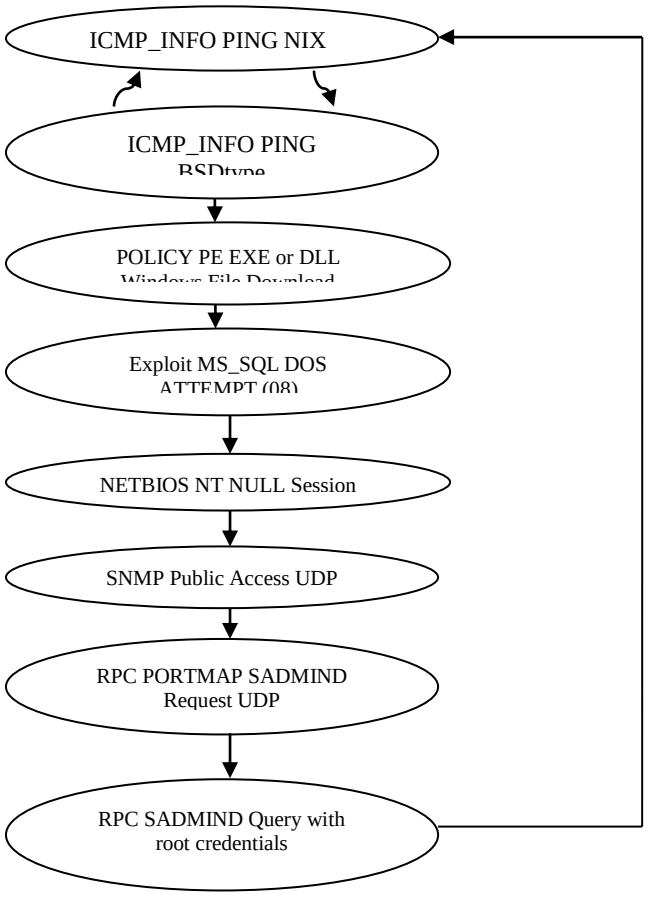

Figure 5: Attack Graphs for the MIT Lincoln LLDOS 1.0 Attack Scenario

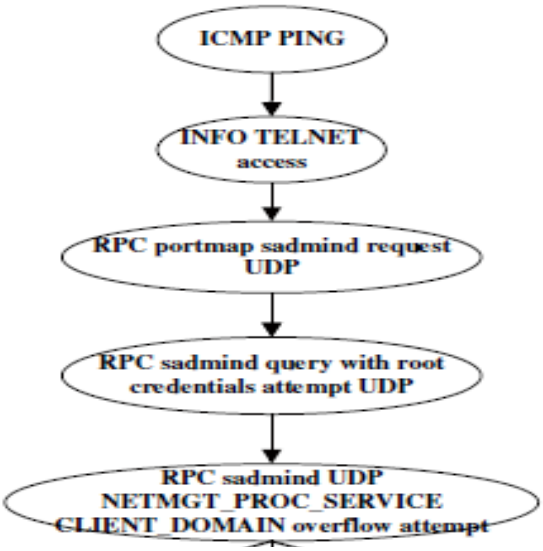

Figure 6: Exploit Oriented Graph (Adapted from Li *et al.*, 2007)

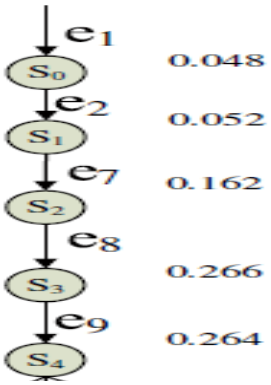

Figure 7: Attack Graphs with Predictability Value (Adapted from Li *et al.*, 2007)

4. CONCLUSION

The Threat Prediction Model outperforms the existing threat prediction model. Although, we have only compared the Threat Prediction Model with Li *et al.*, 2007; however, the model has potential of outperforming other models, which we have not compared with it because no study have been conducted on their ability to predict complex attack scenario. The result shows that Internet-facilitated Threat is recognized better using Actionable Sequential Association Data Mining. It also proves that the use of actionable attributes and context enhances Data Mining for attack prediction and threat strategy recognition.

In future, the potential of Evolutionary Techniques and Neural Networks will be explored in determining the window size and selecting the maximum sequence.

REFERENCE

Farhadi H., AmirHaeri M., and Khansari M. 2011. Alert Correlation and Prediction Using Data Mining andHMM. The ISC Int'l Journal of Information Security. July 2011, Volume 3, Number 2 pp. 77-101 Retrieved 13th March 2012 from http://www.isecure-journal.org

Fayyad U., Shapiro G. P., and Smyth P. From data mining to knowledge discovery in databases. AI Magazine, 17(3):37-54, Fall 1996. Retrieved 2nd April, 2014 from http://citeseer.ist.psu.edu/fayyad96from.html

Haslum K. 2010. Real-time network intrusion prevention. Doctoral theses at NTNU, 2010:168.

Jemili F., Zaghdoud M. and Ahmed M.B. 2009. (IJCSIS) International Journal of Computer Science and Information Security, Vol. 5, No.1, 2009.

Katipally R., Cui X. and Yang L. 2010. Multi stage attack Detection system for Network Administrators using Data Mining.

Li W. and Tian S. 2010. An ontology-based intrusion alerts correlation system. Expert Systems with Applications 37:7138–7146.

Li Z., Lei J., Wang L., and Li D. 2007. A Data Mining Approach to Generating Network Attack Graph for Intrusion Prediction. Computer Communications 29.

Maimon O. and Rokach L. 2009. Introduction to Knowledge Discovery in Databases. Retrieved 4th April, 2014 from www.ise.bgu.ac.il

Nanda S. and Deo N. 2007. A Highly Scalable Model for Network Attack Identification and Path Prediction.

Wang L., Liu A. and Jajodia S. Using attack graphs for correlating, hypothesizing, and predicting intrusion alerts. Computer Communications 29 (2006) 2917–2933.

Zaiane O.R .1999. Principle of Knowledge Discovery in Databases. University of Alberta. Department of Computer Science. CMPUT690.

Zhu B. and Ghorbani A.A. 2006. Alert Correlation for Extracting Attack Strategies. International Journal of Network Security, Vol.3, No.3, PP.244–258