IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 11, November 2016

An Online Malicious Attack Detection using Honey Pot and Episode Mining

M. Narmatha[1], Shri Hari Aravind .K[2]

Assistant Professor, Department of Computer Science, Sri Jayendra Saraswathy Maha Vidhyalaya College of Arts and Science, Coimbatore, Tamil Nadu, India[1]

M.Phil Scholar, Department of Computer Science, Sri Jayendra Saraswathy Maha Vidhyalaya College of Arts and Science, Coimbatore, Tamil Nadu, India[2]

Abstract: Internet security is the major part of current network scenario, there are several types of security threats threatens the internet transactions. Even though there are several techniques and approaches are proposed, still some new issues grow tremendously every day. Recently honeypot systems are deployed to trap and trace internet attacks. But the main drawback in it is it accumulates huge size of traced data. The huge data size is very difficult to handle by the network controller. So, effective pruning and summarization of intruder activity is necessary. The proposed system FEM (Fast Episode Mining) blends a new episode mining, pruning, summarizing and allows a network controller to spot malicious activities. So it reduces the time and energy on tackling those huge data's. The new and enhanced attack episode is composed of a series of proceedings. Through these set of events, the intrusion will be detected. This paper focuses on discovering attack episodes for the **Common Internet File System (CIFS) / Server Message Block (SMB),** which is an application layer protocol. The proposed system is designed to effectively locate the suspicious events and proceedings that are very likely a new one, from an immense amount of logged data. The proposed system is based on the SMB with intrusion detection and response, so this is named as SSMB (Secure Server Message Block). In addition the proposed system performs the intrusion response for the specified type of attack. The detected attack will be responded according to the response dataset from the intrusion response tree.

Keywords: Malicious attack, Honey pot, Intrusion Detection System, Episode Mining, Pruning.

I. INTRODUCTION

Information Technology revolution had a great impact on the online applications. It is always considered as a major challenge to most applications. To ensure the security of the online/web information is extremely important. There are several online attacks [1] [2] threatens the current online applications. The main aim of information security domain is to protect, detect and thwart data from corruption, modification, tampering and access [3].

In paper [4] the different types of attacks and its countermeasures are described. Online data security threats are relentlessly inventive. There are several security threats threatens the current internet application and users. Using new ways of annoying activities, the attacker can steal and harm the data [5].

This type of threat is an event that can take advantage of vulnerability and cause a negative impact on the system. The dangerous and potential threats in such scenario should be identified and prevented earlier, and the related vulnerabilities should be predicted to minimize the risk of the security hazard. The proposed system designed with the aim of providing effective counter measures, effective security and tracking intruders in the network. This aims to reduce the data corruption and data access problems and thwarting different online attacks.

The system only requires less iterations and less effort to detect and prevent intruders, this also aim to achieve reliability and security by applying effective pruning techniques in the intrusion log data.

The system aims to provide the following advantages

- Scalability
- Easy accessibility
- Reliability
- Effective IDS helps to detect, trace and counter measure intruders.

The output of the proposed system will prove the enhanced security level. This paper gives the effective IDS log pruning and counter measure selection for intrusions. While a honeypot deployed on the Internet, the can appeal to a lot of security issues, it cannot identify relationships among events to derive attack episodes.

Usually the honeypot log files are accumulating very large set of logs [6]. It is almost impossible for an administrator to identify novel attacks using these massive amounts of log files. The existing system on this issue applies serial episode mining and two-round pruning to efficiently help the administrator identify suspected attack episodes easily from numerous size of data.

DOI 10.17148/IJARCCE.2016.51175

In other words, the system presents likely unknown attack episodes extracted from a honey pot's log files [7][8] to the administrator, allowing him to make decisions, instead of making decisions for the administrator. Once the suspect episodes have been determined as attacks by the authority, the system will log the newly identified attacks in a database so that the administrator does not need to verify the attacks when they next occur. The decision making process in the existing system was performed by the administrator, effective pruning and counter measure selection is become important.

II. PROPOSED SYSTEM

Network intrusion detection and counter measure selection techniques are implemented in earlier stages, but handling huge dataset and selection of appropriated decisions are still a challenging task. The paper introduced a new collaborative technique to detect, thwart and respond network intrusions from the huge honeypot logs.

The aim of the present work is therefore to propose and experimentally evaluate an automated system with use of episode mining and pruning process and effective counter measure generation with user constrains, which able to filter unwanted messages and illegal requests in large scale networks, The Proposed system exploits FEM (Fast Episode Mining) techniques to monitor the continuous events and episodes and cluster the each and every user actions, and also keep the summary about the frequent and anomaly episodes.
This FEM technique utilizes some data mining technique to detect the unusual behavior easily. The next part of the proposed system is to create a new honeypot mechanism with appropriate response process is introduced. This framework is named as SSMB (Secure Server Message Block).

A. Contributions of the Paper

Internet security is the major part of current network scenario, there are several types of security threats threatens the internet transactions. Even though there are several techniques and approaches are proposed, still some new issues grow tremendously every day. Recently honeypot systems are deployed to trap and trace internet attacks. But the main drawback in it is it accumulates huge size of traced data. The huge data size is very difficult to handle by the network controller.

So, effective pruning and summarization of intruder activity is necessary. The proposed system blends a new episode mining, pruning, summarizing and allows a network controller to spot malicious activities rapidly. So it reduces the time and energy on tackling those huge data's.

- The new and enhanced attack episode is composed of a series of proceedings. Through these set of events, the intrusion will be detected. This is performed by applying FEM technique.
- In this paper, we focused on discovering attack episodes for the **Common Internet File System (CIFS) / Server Message Block (SMB),** which is an application layer protocol.
- The proposed system is designed to effectively locate the suspicious events and proceedings that are very likely a new one, from an immense amount of logged data.
- The proposed system is based on the SMB with intrusion detection and response, so this is named as SSMB (Secure Server Message Block)

In addition the proposed system performs the intrusion response for the specified type of attack. The detected attack will be responded according to the response dataset from the intrusion response tree.

B. FEM (Fast Episode Mining):

i. Monitoring:

The first process in the attacker detection process is the monitoring process, where every node will be monitored.

The proposed system creates a secure server response system, which facilitates continues monitoring and controlling network anomalies and intruders.

ii. Event Collection:

After the continuous monitoring process, events are collected and stored in the log. The events are set of actions, which collected from the IDS. The followings are the list of events with event id.

Table 1.0 event description table

Event Id	Event Name
C	Server connection establishment
S	Session creation
N	New user registration process
L	Initial Authentication process
B	Second level authentication process
A	Authentication failed process
R	Resource selection process
D	Network discovery process
M	Monitoring
N	Negotiation protocol
W	Write process
J	Read process

The table 1.0shows the events and its identification number used in the implementation. This contains a set of events and its id.

The every event is created according to the implementation; the above mentioned events are sample event ids used in the proposed system. The system monitors every event and converts into the event id and stores in the database log. This log will be used by the FEM process.

DOI 10.17148/IJARCCE.2016.51175

iii. Correlating:
Correlation is the process of log aggregation and finding association between events, where the events are collected from different sources and that will be integrated as a whole file. In this case, data mining techniques are used.

Initially this correlates and prunes the frequent and non-malicious events and converts into the priority based episodes. The episodes are the collection of events, which has been collected at every specific time period.

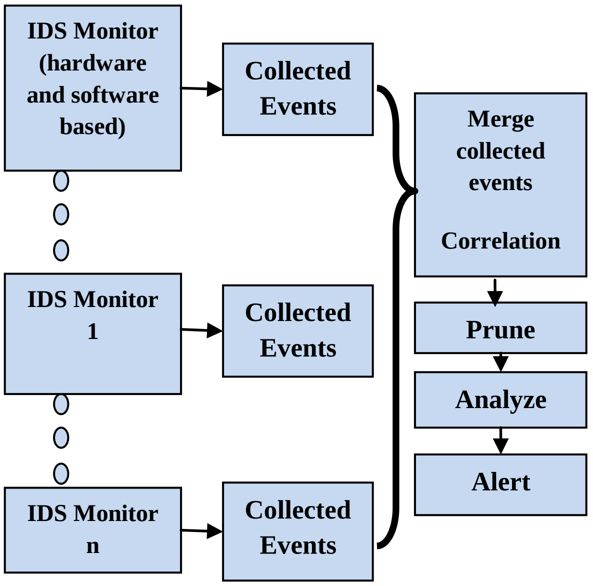


Fig 1.0 over all process of the proposed system

The above fig 1.0 represents the event collection, merge, correlation and analysis process. Here correlation and aggregation functions are used. A **correlation function** is a function that gives the statistical correlation between random variables, contingent on the spatial or temporal distance between those events.

If one considers the correlation function between events representing the same quantity measured at two different points then this is often referred to as an autocorrelation function, which is made up of autocorrelations.

For possibly distinct two events E(x) and E(y) at different point's x and y of some space, the correlation function is

$$C(x,y)=corr(E(x),E(y)),$$

Where, corr is described in the above formula is represents the process of correlation. In this definition, it has been assumed that the stochastic variables are scalar-valued. If they are not, then more complicated correlation functions can be defined.

iv. Pruning:
After the aggregation, correlation processes, the system performs the pruning process, where irrelevant and unwanted episodes are pruned.

v. Episodes:
The collection of events are listed based on the time is called as an episode. The serial process of a client is listed below. The list of processes and the event number are collected sequentially.

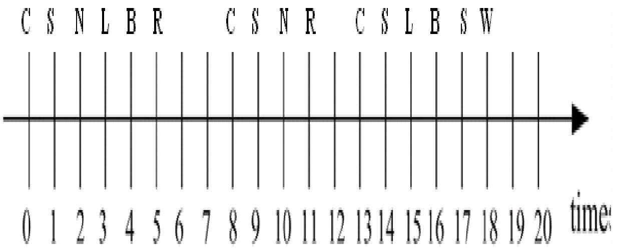


Fig 2.0 serial episodes

From the collected and potential episodes from honeypot log files, the analysis is performed. This is very difficult to analyze the whole honeypot log file. So, effective pruning techniques are used. Here the fig 3.2 shows the episodes, which is a number of events within an event sequence.

In serial episode mining, data are regarded as a sequence of events, where each event has an associated time of occurrence, and thus offers significant use in identifying possible Internet intrusions.

Fig. 3.2 is an example for illustration. Suppose that (C, S, R) is a serial episode, which means that after event C, event S occurs, and the episode ends with event R. When considering events and their time of occurrence, possible patterns are {(C, 0), (N, 2), (B,4)}, {(C, 0), (N, 3), (R,5)}, {(C, 8), (N, 10), (S,9)}, {(C, 13), (S, 14), (B,16)}, etc. Serial episodes are of interest to this paper as hacker attack actions also have time sequences.

Let e-type = {C, S, N, L . . .} be a collection of all event types in the event sequence. An event (A, t) denotes an A type event occurring at time t.
An event sequence s is associated with two times: the starting and ending time, denoted as (s, Ts, Te) which, more precisely, can be expressed as {(A1, t1), (A2, t2), (An, tn)}. An episode is a number of events within an event sequence. The length of an episode is the number of events in the episode.

vi. Frequent Episode Mining:
The system utilizes the FEM (Fast Episode Mining), algorithm to prune unwanted episodes. The fast episode mining utilizes the data mining technique to mine effective serial episodes.

vii. Candidate Generation:
The candidate generation is the process of extracting list of items and item sets from the huge dataset. In this project, the collected events and its event list are used for candidate generation process. The following is the sample of candidate generation listed in table 2.0.

Table 2.0 support calculation of event type

Event id		Candidate generation
E	A	AB
S	B	AC
L	C	AE
A	E	AL
B	L	AS
C	S	ABC,AEL,CSL,ELS,CBA, etc.,

viii. Frequent episode detection:

The frequent episode detection is the process of identifying the total number of occurrences of an event with various time intervals.

For example, the log from every time period is collected as an event and that will be applied to find the frequency (1).

$$\text{Support }(E)=(T(_{E}))/N \qquad (1)$$

Where E is an event, $T(_{E})$ is the total number of occurrences of E and N is the total events in the episodes. In the proposed system the honeypot data's are collected and applied for frequent episode detection.

Table 3.0support calculation of event type

Tid	Events	Occurrences/total transaction
1	C and S,	Total support : 6 Support of(C)=(4/6)=66.6% Support of(S)=(5/6)=83.3% Support of(L)=(1/6)=16.6% Support of(C, S)=(3/6)=50.0%
2	L and S	
3	C	
4	C and S,	
5	C and S	
6	S	

In order to identify the frequent episode different events are considered and applied into the formula (1), thus the final output will be applied into the new honeypot technique.

As like the support additionally, the confidence among different events is considered by the following formula (2).

$$\text{Confidence }(E\rightarrow E1)=P(E/E1) \ (2)$$

Where E is an event, E1 is the total number of occurrences of E1and Eis the total number of occurrences of event 1 is the total transaction in the dataset. In the proposed system the honeypot data's are collected and applied for frequent episode detection.

The rule generation from the confidence is important to analyze the association between events. X→ Y holds with confidence c if c% of the transactions in D that contain X also contain Y . Rules that have a c greater than a user-specified confidence is said to have minimum confidence.

Table 4.0 support calculation of event type

Tid	Events(C,S)	Given C→ S Confidence=Occurrences of (C)/ Occurrences of (S)
1	C, F	Total support : 6 C → L,F (1/1)=100% L and S → B =(2/4)=50% S → B =(3/4)=75%
2	L and S, B	
3	S, B	
4	F and S, B	
5	C, B	
6	S, B	

In order to identify the frequent episode different events are considered and applied into the formula (2) named as confidence, thus the final output will be applied into the next process.

Algorithm: FEM
Input: events from honeypot log
Output: Frequent Episodes
Steps:

```
1.   L==0, let L is denoted as an empty dataset
2.   Perform candidate generation as in table 3.2
3.   Find support S and confidence C
4.   For each event E in event log
5.   If(S(E)>min_sup)
6.   If(C(E)>min_conf)
a.   L U S(E)
b.   L++
7.   Desc(L)
8.   Get FEM=top(k) items form L.
9.   Return  FEM
10.  end
```

Honeypot data logs usually accumulate very quickly, and thus cannot be manually analyzed by administrators. So the above steps mine both the frequent and infrequent episodes and helps to take necessary actions according to that. This study aims to identify suspected attack episodes from the large amount of raw data in honeypot logs. Administrators can thus focus only on these selected episodes and make decisions based on their expertise, instead of reading the logs to find intrusions.

ix. Analysis

The proposed system takes the codes in the command fields of client connection packets, also known as events, for analysis. In other words, the system only mines significant misused connections contrary to the normal client connection process. The administrator needs to read more detailed information from the honeypot logs or firewall logs to make decisions according to his knowledge. In fact, no method can exactly and automatically identify novel attacks without human assistance. But in the proposed system, the system finds the behavior and makes the event type. Once a suspected

DOI 10.17148/IJARCCE.2016.51175

episode has been determined as an attack by the administrator, the attack episode will be stored in the database with the id of "A" so that the administrator will not have to check it again when it next occurs.

After the successful analysis, the system performs the counter measure selection process. This needs an effective alert mechanism to select optimal solution for the detected attack.

III. RESULTS AND DISCUSSION

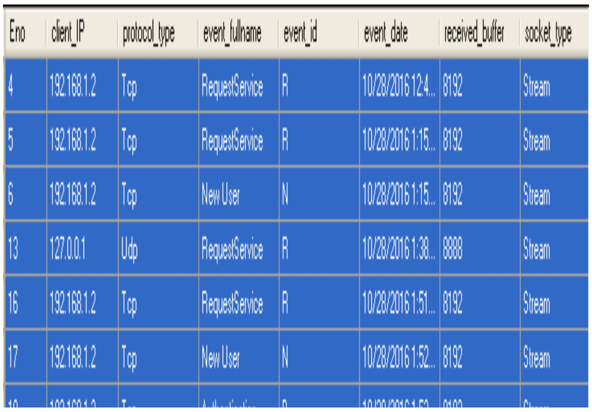
A. Test Samples:

The system uses the following test interface. The system has constructed with the client server based approach. The server has the monitoring and authentication criteria providing processes.

In the proposed system, a client node sends out a request message to server. The server will receive all details of the client

The system presents two methods to generate authentication criteria's according to the network behavior purposes. An important characteristic of client authentication criteria is that the amount of computation needed to resolve it can be estimated fairly well.

Note that the authentication criteria used in the defense against online malicious attacks need not require naturally sequential operations. It is important that a data cannot be accessed by anyone until a pre-determined authentication is solved.



Eno	client_IP	protocol_type	event_fullname	event_id	event_date	received_buffer	socket_type
4	192.168.1.2	Tcp	RequestService	R	10/28/2016 12:4...	8192	Stream
5	192.168.1.2	Tcp	RequestService	R	10/28/2016 1:15...	8192	Stream
6	192.168.1.2	Tcp	New User	N	10/28/2016 1:15...	8192	Stream
13	127.0.0.1	Udp	RequestService	R	10/28/2016 1:38...	8888	Stream
16	192.168.1.2	Tcp	RequestService	R	10/28/2016 1:51...	8192	Stream
17	192.168.1.2	Tcp	New User	N	10/28/2016 1:52...	8192	Stream

Fig 2.0 event table

The fig 2.0 represents the collected events from the honeypot system. The server will randomly selects some images and split into several portions. The portions should be monitored with every step.

B. **Episode pruning Time:**

Finding frequent episodes and the frequent abnormal event form the client IP are collected in this phase. This episode also contains the behavior score of the client. this score determines the counter measure selection.

C. Attack event "A" Identification:

The following chart represents the time variance between the normal user and attacker. If the attacker tries to solve the game by taking more counts and time. The system will find out the time variance in order to detect the attacker.

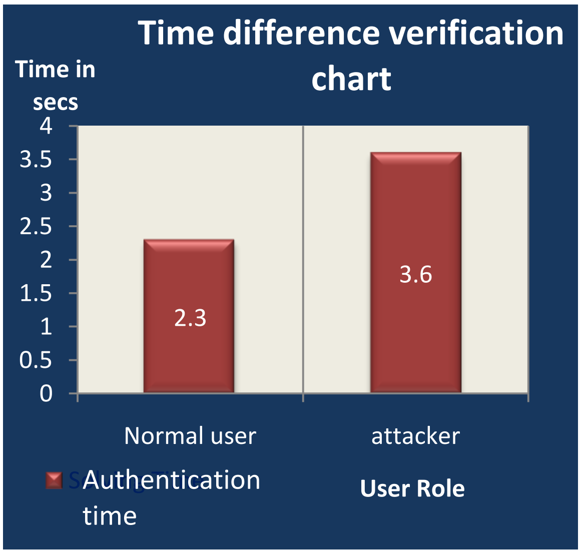


Fig 3.0 attacker event detection using authentication delay

The server will maintain the time and clicks of the given authentication criteria. The time and clicks may vary based on the user and attacker. The system will effectively find the attack with minimum time span.

IV. COMPARTIVE STUDY

In proposed system, the comparison process consists with two performance metrics: detection accuracy and Latency. Latency is the time taken for every process such as event collection time, attacker event detection time, pruning time and counter measure selection time.

This section performs the comparative study between existing system and the proposed system. This has two types of existing system based on the performance. One is based on the pruning technique another one is the traditional honeypot system to handle intrusions.

1. Episode Pruning Techniques:

- Existing Redundant/Correlated Episode Pruning (RCEP).
- Proposed FEM

2. Honeypot based techniques for Intrusion detection and response:

- Existing automated response system (ARS)
- Proposed SSMB- dynamic counter measure selection based on the behavior score.

The comparison between the above two techniques are compared by the following section.

DOI 10.17148/IJARCCE.2016.51175

Pruning Delay Comparison for Single Set Event

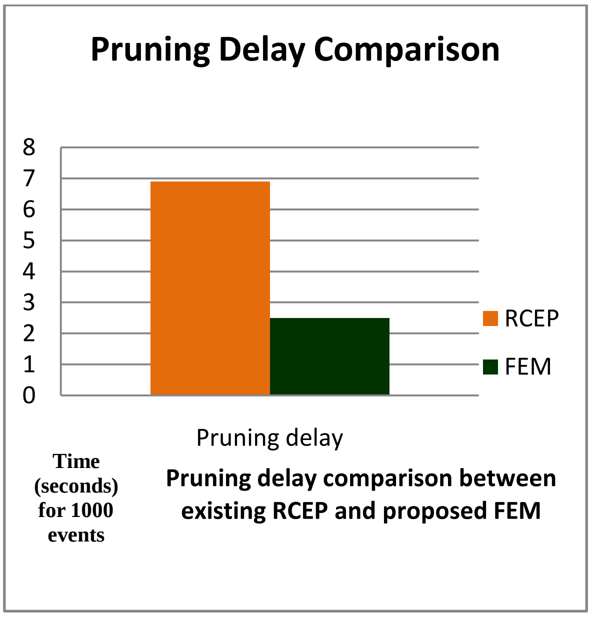


Fig 4.0 pruning delay comparison chart

As per implementation with different datasets and analysis the fig 4.0 shows the difference between the existing and the proposed techniques based on the pruning delay. Achieving these performance benefit in the domain of server client request Filtering concept is not a small task, even the load has increased the performance will be effectively analyzed.

V. CONCLUSION

The system proposed a new honeypot based defending system against intrusions. The system has created new IDS with customized and dynamic counter measure selection. The presented FEM and SSMB based intrusion detection and response algorithm identifies the malicious events from the event log rapidly and performs optimal countermeasure selection. This has the ability to detect and mitigate online malicious attacks in the distributed network environment. SSMB exploits the active attack table and user score creation model to conduct attack detection, prediction and response. The proposed solution utilizes the decentralized distributed intrusion detection and response to improve the reliability, detection accuracy and defeat victim exploitation phases of collaborative attacks in network environment.

The proposed protocol improves the detection accuracy by implementing effective pruning techniques and shows that the proposed solution can significantly reduce the risk of the network system from being exploited and abused by internal and external attackers. Proposed system investigates both the network and host based IDS approach to counter different types of online malicious attacks. To improve the detection accuracy the system performs active attack table and evernt and episode verification schemes.

REFERENCES

[1] Liang, Yingbin, and H. Vincent Poor. "Information theoretic security."Foundations and Trends in Communications and Information Theory 5.4–5 (2009): 355-580.

[2] Canali, Davide, and Davide Balzarotti. "Behind the scenes of online attacks: an analysis of exploitation behaviors on the web." 20th Annual Network & Distributed System Security Symposium (NDSS 2013). 2013.

[3] Bascle, Jeff P., et al. "System and method for reducing the vulnerability of a computer network to virus threats." U.S. Patent No. 7,571,483. 4 Aug. 2009.

[4] Kirda, Engin, et al. "Behavior-based Spyware Detection." Usenix Security. Vol. 6. 2006.

[5] Dai, Shuaifu, et al. "A framework to eliminate backdoors from response-computable authentication." 2012 IEEE Symposium on Security and Privacy. IEEE, 2012.

[6] Thakar, Urjita, Sudarshan Varma, and A. K. Ramani. "HoneyAnalyzer–analysis and extraction of intrusion detection patterns & signatures using honeypot." Proceedings of the Second International Conference on Innovations in Information Technology. 2005.

[7] Visoottiviseth, Vasaka, et al. "Distributed honeypot log management and visualization of attacker geographical distribution." Computer Science and Software Engineering (JCSSE), 2011 Eighth International Joint Conference on. IEEE, 2011.

[8] LI, Jing, Yong SHI, and Zhi XUE. "Initiative Defense Research Based on Log Analysis of Honeypot [J]." Information Security and Communications Privacy3 (2009): 039.

DOI 10.17148/IJARCCE.2016.51175