Developing a Host Intrusion Prevention System by Using Data Mining

Prof. Dr. Alaa Al-hamami & Tahani Alawneh
College of Computer Sciences and Informatics
Amman Arab University
alaa_hamami@yahoo.com tahani_alawneh@yahoo.com

Abstract

*Intrusion Prevention Systems (**IPS**) is the most important solution for providing a high level of security all over the networks today. **IPS** is evolving recently in a way that is expected eventually to replace other security solutions such as firewalls and anti-viruses. To over come the static signature detecting mechanism to identify intruders that exists in all host based IPSs which in turn needs to be updated from time to time to insure the most accurate detection. In this paper we introduce a four tier host based **IPS** that uses data mining technique, namely decision tree, as a detecting mechanism. The input parameters for the prior decision tree algorithm are the most infected or targeted computer resources by intruders, instead of a static signature database. Three test scenarios were performed to investigate the ability of the proposed **IPS** to classify intruders correctly.*

***Keywords**: Intrusion prevention system, data mining, decision tree, intruder, and information security.*

1. Introduction

Security risks come in the forms of Viruses, Trojan Horses, Malwares, Application security holes, or a Hacker or a Cracker who tries to mess up with your data. 80 percent of security breaks comes from internal intruders [1]. Since 1995, the annual increase in risk from internet hacking is up 60% per year, while the annual increase in risk from viruses and worms is up over 100% per year [3].

Accordingly, there are many solutions to stand for such intruders, such as firewalls, anti-viruses and **IPS**s. Among all security solutions, **IPS** has its own special characteristics in analyzing, detecting and preventing intruders' acts; this paper aims to enhance intruders' detection, by using data mining approaches, namely decision tree. Data mining is the analysis of (often large) observational data sets to find unsuspected relationships, and to summarize the data in novel ways that are both understandable and useful to the data owner [4].

The input for the decision tree algorithm is a comma separator file format that has entries for the most infected or targeted resources by intruders, whether an intruder was a Virus (a type of malicious software that can destroy the computer's hard drive, files, and programs in memory, and that replicates itself to other disks), a Trojan horse (computer program that is apparently or actually useful and contains a harmful code), a Spayware (a program that collects data about your **PC** or network and sends it to another destination), or even Human acts. This file was pre-populated with entries that indicate the most famous intrusion infection symptoms. But it keeps growing each time an anomaly in one of these resources is being detected. With decision tree, the file is being treated to decide whether the resource behavior was due to a normal action or an intruder action. The user is informed, and chose to commit the decision or not, at that time the file is being updated depending on the user response, which provides a feedback to the whole process and definitely causes future inspections to be more realistic. WEKA workbench is used as a data mining tool in this paper.

2. Data Mining and IPS

Data mining involves the use of data analysis to discover previously unknown, valid patterns and relationships in large data sets. These tools can include statistical models, mathematical algorithms, and machine learning methods. Recently data mining started to enter information security field such as intrusion detection [5].

WEKA (Waikato) Environment for Knowledge Analysis) is a collection of machine learning algorithms for data mining tasks implemented in Java language, it contains tools for classification, regression, clustering, association rules, and data visualization.

Intrusion prevention is defined as the process of intelligently monitoring the events occurring in a computer system or a network, analyzing them for signs of violations of the security policy and take appropriate actions. Many researches demonstrated that data mining techniques such as decision trees,

978-0-7695-4959-0/13 $25.00 © 2013 IEEE
DOI 10.1109/ACSAT.2012.103

association rules, clustering and others are well known, and used for vulnerabilities detection.

One paper used the association rule to identify anomalies depending on system patterns, the paper associates processes with their resources to decide such anomalies [6]. Another paper characterized normal system activities in a profile, and processed it by fuzzy data mining algorithms to decide abnormality, and take the appropriate prevention action accordingly [18]. A third one stores events related to the system and the network that will be treated later by the α-algorithm to decide and prevent intruders [7].

This research uses system resources as an input attributes for the decision tree algorithm of WEKA tool, it depends on system alerts to report for any odd behavior in the system. This study does not relate resources to the user or to a specific process, like many previous papers; indeed it relates resources to a pre defined baseline measure for each resource. This procedure is actually too close to the previous researches, in the idea of finding a real measurements or indicators for anomalies from the system itself, but this study still has some distinguish technique, that plays a key role in decreasing false positive alarms and false negative alarms, by providing a simple feedback to commit the action taken by **IPS** in order to tune it correctly with time.

The WEKA implementation of the used decision tree algorithm in this study is **J48**. This algorithm was developed and refined over many years by J. Ross Quinlan of the University of Sydney.

3. The Proposed IPS Architecture

The system was designed in the form of four logical tiers; each tier plays a certain role. And these tires integrate with each others to produce the whole complete system, eventually. A specified technology was used in each tier when developing the system, in order to meet its' rule.

The first tier is the Client tier; it contains the user interface controls, which in turn takes the role of communicating with all the code behind. VB.Net technology was used as a developing environment tool for this tier.

Then Comes the Business Logic Services tier, this tier will handle data mining algorithm and hands the output to Data Services tier. WEKA the famous data mining tool was used to perform **J48** association algorithm on the input data at this tier.

The Data Services Tier is the core of the proposed **IPS** which has two jobs indeed; it takes the appropriate action when ever an intruder is being detected, and alarms the user about the suspected action. The first job is the responsibility of the Prevention Module of that tier. While the second job, is the responsibility of the Alarm Module. This module depends on Business Logic Services tiers' decisions to start its' job. ASP.Net technology was used as a developing environment tool at this tier.

Finally, the Data tier contains the input data for the Business Logic Services tier, which is populated through system alerts. It is being updated through a feedback sent from Data Service Tier. Comma separator file format is being used at that tier. Figure1 demonstrates system architecture.

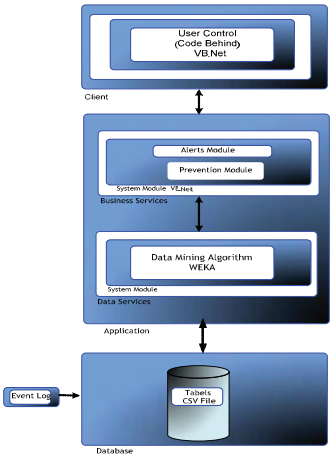

Figure 1. System Architecture

3.1. Data Tier

To build the lowest logical tier (database tier) that appears in figure 1, the following steps were performed:

Specify system resources that to be monitored.

After reviewing the most famous attacks (security holes, viruses, Trojan horses, warms, spaywors …) targeting computers in the last five years, certain symptoms were noticed in common, that indicate an up normality in the performance of the infected machine. This gave the idea to monitor the most attacked resources in computers and take an action once an anomaly behavior is noticed in any.

Monitor the performance of selected resources for a period of time that would reflect the normal operation (threshold) of each.

The previous resources are being monitored for a period that is quite enough to reflect system normal operation (Finding the baseline for each resource).

Build system alerts based on the previous thresholds.

When ever a resource exceeds its' threshold, a system alert would be triggered which in turn will write an alarm tag to the related resource field in the input data file. Figure 2 shows a portion of the built input data file.

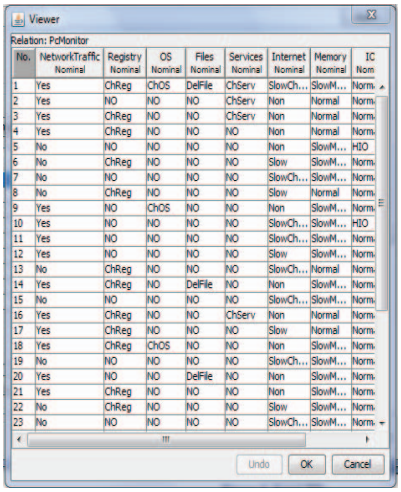

Figure 2. Input file

3.2. Data Service Tier

The input data file is being treated by WEKA decision tree algorithm **J48**, Figure 3 shows the predictive part of the output.

```
===Stratified cross-validation===
===Summary===
Correctly Classified Instances          25
80.6452%
Incorrectly Classified Instances         6
19.3548%
Confusion Matrix===
 a  b   <-- classified as
 | 0  25 a = Int
 | 0  6  b = App
```

Figure 3. Predictive part of J48 output

This part gives an estimate of the trees predictive performance. It was obtained using cross validation of 10 folds. As it is shown about 19% of the instances (6 out of 25) has been misclassified in the cross validation, and the confusion matrix at the end it shows that none of the Int (intrusion) class were classified incorrectly, but 6 of the App (Application) instances was classified as an Intrusion. This means that the J48 algorithm will predict attacks with accuracy of 80%. This will be tuned with time as the input file grows, and through the feed back of the user.

3.3. Business Service Tier

Each system alert takes a certain action to prevent the intruder. This action is programmed using VB.net. Add to that, this action is also committed in the input file in order to reflect the actual case of the system, since the alert could be a false positive or a false negative one. Table 1 shows a portion of these actions.

Table 1. IPS actions

Resurce	VB file	Written output Value	Action Taken
NetworkTraffic	NetworkTraffic.exe	High	Alarm the user & Reset Connection
		Normal	
		UpNormal	Alarm the user & Reset Connection
Registry	Registry.exe	ChReg	Alarm the user
		No	
OS	OS.exe	ChOS	Alarm the user and restart
		No	

Figure 4, shows the graphical user interface alarm message of the **IPS** related to the Network Traffic entry in table 1, which will appear to the user when an anomaly in the network traffic is being detected. The same message will appear for any monitored resource exceeds the normal operation threshold.

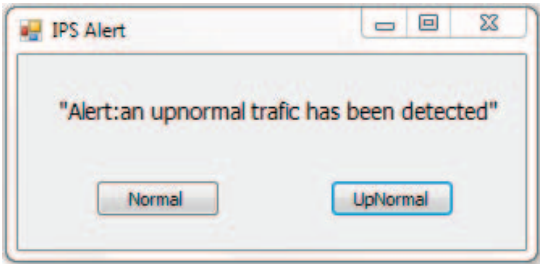

Figure 4. IPS network alarm

3.4. Client Tier

Whenever an alert appears as shown in Figure 4, it asks the user through two buttons to commit this alarm as a normal behavior or an anomaly. This will be written back to the input database. As mentioned before, this will reflect the actual case of the system, since the alert could be a false positive or false negative one. This will help to replaces wrong alarms decided by the proposed **IPS** (false positive) with the right ones, which in turn will push future prediction towards reality.

4. Testing

Three test scenarios were used, in the first one the conficker virus was injected, while boot.ini and desktop.ini were injected in the second scenario, and lastly a network traffic inspection application named Net Flow analyzer was ejected to the system, the result was as the following:

4.1. Applying conficker virus

Table 2 shows that, the value for Correctly Classified Instances of the input data file before applying the test was 78.125% . After applying the test the following happened:

Table 2. J48 output for conficker

Viruses	Triggered Alarms	Previous Correctly Classified Instances	Correctly Classified Instances	Committed Correctly Classified Instances
conficker	High Traffic & service change	78.125%	80.6452%	81.25 %

1. Two alarms on the system: High Traffic and Service Change.
2. The output for the J48 algorithm had the following value for Correctly Classified Instances = 80.6452**%**

As shown in Table 2 the Correctly Classified Instances value is 80.6452% , which indicates that the action was classified as intrusion. Although un-triggered alarms are being represented as missing values in the input data file.

Committing the action back to the input file increased Correctly Classified Instances value about 3.125% , this happened because the last attribute in the file value named Infection is no longer a missing value. As it was before predicting, it was written back that this value is an infection, which caused this increased. This will help in future prediction, as the predicting rate will be more realistic, as False positive and false negative decisions will decrease with time.

4.2. Applying boot.ini / desktio.ini virus

Table 3 shows that, the value for Correctly Classified Instances of the Input data file before applying the testis now 81. 25% . After applying the test the following happened:

Table 3. J48 output boot.ini / desktop.ini

Viruses	Triggered Alarms	Correctly Classified Instances	Committed Correctly Classified Instances	Previous Correctly Classified Instances
Boot.ini Desktop.ini	Slow memory	81.25%	81.25%	81.8182 %

1. Two alarms on the system: Slow memory.
2. The output for the J48 algorithm had the following value for Correctly Classified Instances = 81.25%, which indicates an infection.

Committing the action back to the input file increased Correctly Classified Instances value about 0.5682%, this happened because the last attribute in input file value named Infection is no longer a missing value. As it was before predicting, it was written back that this value is an infection, which caused this increased. This will help in future prediction, as the predicting rate will be more realistic, as False positive and false negative decisions will decrease with time.

4.3. Applying net flow analyzer

Table 4 shows that, the value for Correctly Classified Instances of the input data file before applying the testis now 81. 8182% . After applying the test the following happened:

Table 4. J48 output for net flow analyzer

Viruses	Triggered Alarms	Correctly Classified Instances	Committed Correctly Classified Instances	Previous Correctly Classified Instances
Boot.ini Desktop.ini	Slow memory	81.8182 %	81.8182 %	79.4118 %

1. One alarm on the system: Slow memory.
2. The output for the J48 algorithm had the following value for Correctly Classified Instances = 81.8182 %, which indicates an infection.

Committing the action back to the input file decreased Correctly Classified Instances value about 2.4064% , this happened because the last attribute in input file value named Infection is no longer a missing value. As it was before predicting, it was written back that this value is an infection, which caused this increased. This will help in future prediction, as the predicting rate will be more realistic, as False positive and false negative decisions will decrease with time.

5. Conclusions

Using data mining in the detecting mechanism of **IPS**, has directed **IPS** to a new vision of not to depend on a database of attack signatures to inspect intruders, that must be updated eventually.

Using computer system resources as input attributes for data mining techniques, has provided a dynamic real inputs, which is so close to reality, which causes decisions taken by **IPS** to be more objective and trough.

Depending on such resources to be used as input values for IPS, simplified the diction mechanism and made it efficient at the some time. This helps IPS to be spread on the top of each PC in the network, which maximizes security Level.

After applying the three test scenarios explained previously the following conclusions can be figured:

1. Using **J48** decision tree algorithm in **IPS** detecting mechanism did indicate intruders in an acceptable rate.
2. Using system resources, such as memory and network did indicate anomaly acts, depending on deviations in previously measured natural baseline behavior.
3. The percentage of prediction could be tuned to reflect realistic values, by providing a feed back to the same input file through committing **IPS** actions back to the file, and analyze it once again using the same **J48** algorithm.
4. A natural act that triggers an alarm in system resources, due to unexpected use of such resources which happened to be above the natural baseline, was treated as an intrusion and this is called the false positive alarm. This is healthy, and could be treated in two ways: either to tune the baseline of the related resource or depending on the feedback of the IPS that will commit the wrong alarm in the dataset as a normal application, which causes the correctly classified instances rate to decrease. This feedback with the enough sufficient time will tune the system by itself.
5. As the input file grows up, the percentage of prediction will be more representative to reality.

References

[1] Alaa H Al-Hamami and Saed Al-Ani, Technology of Information Security and Protection Systems, Al-Awael, for Publishing and Distribution, 2007.

[2] Alaa H Al-Hamami, Datamining: Concepts, Techniques and Applications, Al-Ithraa for Publishing and distribution, 2007.

[3] Daniel T. Larose, Data Mining Methods And models, Bilkent , Central Connecticut State University, Wiley Interscience , 2006.

[4] W.lee, and S.Stolfo, . Data Mining Approaches For Intrusion Detection Models, 2004.

[5] Wenke Lee, and Salvatore J. Stolfo, Data Mining Approaches for Intrusion Detection, Columbia University, 2000.

[6] Theodoros Lappas,, and Konstantinos Pelechrinis, Data Mining Techniques for (Network) Intrusion Detection Systems, UC Riverside , 2009.

[7] Theodoros Lappas,, and Konstantinos Pelechrinis, The CISSP Prep Guide: Golden Eddition, UC Riverside, 2009.

[8] A.B. Smith, C.D. Jones, and E.F. Roberts, "Article Title", Journal, Publisher, Location, Date, pp. 1-10.

[9] Andre' Muscat, , A Log Analysis based Intrusion Detection System for the creation of a Specification Based Intrusion Prevention System, University Of Malta, 2003.

[10] Anshu Veda, Prajakata Kalekar, and Anirudha Bodhankar, Intrusion Detection Using Data Mining, 2001.

[11] Cumhur Doruk Bozagac, Application of Data Mining based Malicious Code Detection Techniques for Detecting new Spyware, Bilkent University, 2005.

[12] Daniel T. Larose, Discovering Knowledge In Data: An Introduction to Data Mining, Central Connecticut State University, Wiley Interscience , 2005.

[13] Daniela Schiopu, and Irina Tudor, Analyzing Information Security Issues Using Data Mining Techniques, University of Ploiesti, 2008.

[14] Earl Carter, and Jonathon Hogue, Intrusion Prevention Fundamentals, Cisco Press, 2006.

[15] Harold Tipton, and Micki Krause, Information Security Management Handbook, 5th edition, Auerbach Publications, 2004.

[16] Ian Witten, and Eibe Frank, Data Mining Practical Machine Learning Tools and Techniques, 2nd Edition, Elsevier, 2004.

[17] Jiawei Han, and Micheline Kamber, Data Mining: Concepts and Techniques, Simon Fraser University, Morgan Kaufmann Publishers, 2000.

[18] Juan Pablo Pereira, Comparison of Firewall, Intrusion Prevention and Antivirus Technologies, Juniper Networks, 2004.