DEVELOPMENT OF AN ENHANCED MODEL FORINTRUSION DETECTION IN A CLOUD COMPUTING ENVIRONMENT

ABSTRACT

Intrusion is one of the important issues in all networks especially in cloud computing where all the services are served using internet. The fully distributed and open structure of cloud computing and services have made it even more attractive target for potential intruders. As hackers and attackers get sophisticated, the defense to prevent such attacks must be sophisticated as well. A cloud computing system can be exposed to a number of threats, including the integrity, confidentiality and availability of its resources, its data, and the virtualized infrastructure can be vulnerable, and can be used as a launch pad for new technology attacks. The problem becomes even more critical when a cloud with massive computing power and storage capacity is misused by an internal intruder as a malicious party. This research developed an enhanced model for intrusion-detection that monitors and analyzes data in a cloud environment and detect intrusion in the system or network. The model can detect intrusions from external and malicious internal (authorized and unauthorized) users by normalizing and classifying all data packet using machine learning techniques. The developed system is an enhanced model of Zhang by combining it with two machine learning techniques: Support vector machine and Bayesian network to aid in the classification of normal data and intrusion data to detect intrusions. Object oriented analysis and design methodology (OOADM) was used. The developed model is evaluated and found to be able to make strong predictions, detects attacks and still maintain the efficiency of the network. The new system, when implemented, can detect intruders by classification of data packets and also improve on the existing system in terms of providing more accurate and more efficient intrusion detection. It also provides worthwhile information about malicious network traffic, helping to identify the source of the incoming probes or attacks, collecting forensic evidence that can be used to identify intruders and alerting security personnel that a network invasion maybe in progress.

Keywords: Intrusion, detection, Internet, protocol, computing.

1. INTRODUCTION

The cyber security industry is one of the fastest growing industrytoday. Trends such as cloud computing, virtualization, and IoT (Internet of Things) have made data not only the most lucrative asset, but also the most vulnerable. As networks get larger, the attack surface for hackers increases, making cyber risk a prevalent problem. As hackers and attacks get sophisticated, the defense to prevent such attacks must be sophisticated as well.There are many common middle box solutions that exist today. Applications such as firewalls and honeypots are the first line of defense against a cyber-attack. Firewalls act as a filter for network traffic - using a set of rules.A firewall will prevent hosts from outside the internal network to connect to a secure end system. Firewalls suffer from the ability to maintain state; a persistent attacker can easily bypass a firewall and gain entry into the enterprise network. Honeypots act like a trap for attackers - by advertising the possibility that it contains sensitive information; hackers will try to access the honeypot and are then blocked from the network. However, honeypots are only successful if they can bait the attacker. If the attacker realizes that the honeypot is trying to fool them, they can ignore it and continue to attack the network.Thus, a necessity has been created for a system that can learn the structure of network data and differentiate normal from abnormal network traffic. To achieve this, an enhanced model for intrusion-detection in a cloud computing environment has been developed. Intrusion detection system (IDS) is a system that monitors and analyzes data to detect any intrusion in the system or network. An IDS comes in two forms: host-based IDS (HIDS) and network-based IDS (NIDS). A HIDS

is a software that runs on a host machine or on a centralized controller to monitor access to the file system, verify chains of system calls, or malicious changes to environment/system variables. On the other hand, a NIDS monitors traffic that travels through a network and usually runs on edge routers/switches. The greatest challenge that network-based detection systems must overcome is trying to model the behavior of normal and abnormal traffic. Machine learning (ML) has become the greatest trend in technology today.Complex problems that donot seem to have a clear-cut answer can be realized using an ML model. The model will learn the necessary features of a dataset to make strong predictions. Today, cutting-edge research in IDSes employs a plethora of different ML techniques to see if they can be used in an IDS to secure enterprise networks.

2. RELATED WORKS

In the earlier works on intrusion detection system, Hwang *et al*. (2007) proposed a hybrid system that combines a signature-based IDS with an anomaly detection system in a cascade structure, achieving twice the detection accuracy of IDS only system.

Patel *et al.* (2013) also proposed the characteristics of the intrusion detection system for cloud environment. The method or algorithm for such proposal was not proposed by the author. The implementation of the proposed concept was the future scope of the paper.

Kleber *et al.* (2010) have proposed a Hybrid Intrusion Detection System for Cloud and Grid environment. This system can detect any kind of attack; hence the efficiency of detection is very less. This system cannot be deployed into a real time distributed environment, as the system cannot synchronize well with the other Intrusion Detection Systems in the network and the architecture and working of both models is completely different.

Tupakula*et al.* (2011) have proposed a Hybrid Intrusion Detection System for Infrastructure as a Service Cloud. This system cannot handle large scale, dynamic, multithread and data processing environment. Since the system has been proposed for Infrastructure as a Service Cloud, the synchronization character is not applicable to the system. Software as a Service and Platform as a Service are the other two services of cloud, which has not been considered by the authors.

Kholidy*et al.*(2012) have proposed a framework for Intrusion Detection in Cloud Systems. This framework does not detect the intrusion in a faster manner; hence the efficiency of detection is very less. This system can partially only handle large scale, dynamic data which is another drawback of the system. The authors did not narrate the scope for implementing the algorithm for the private cloud environment.

Xinwang*et al.* (2010) has proposed and developed an Intrusion Detection System for cloud with the central management approach. The developed model is not scalable. The efficiency of the system gets reduced when the system is scaled. The implementation of the developed system is quite complicated and difficult to manage.

Roschke*et al*. (2009) proposed an integration solution for central IDS management that can combine and integrate various renowned IDS sensors and output reports on a single interface. The authors have proposed an effective cloud IDS management architecture, which could be monitored and administered by the cloud user. They have provided a central IDS management system based on different sensors using the intrusion detection message exchange format (IDMEF) standard for communication between different IDS sensors.

Irfan and Hussain (2011) proposed an IDS service at cloud middleware layer model, which has an audit system designed to cover attacks that NIDS and HIDS cannot detect. The authors have tested their IDS prototype with the help of simulation and found its performance satisfactory for real-time implementation in a cloud environment. Although the security policies compliance check for cloud service provider and their reporting procedures to cloud users was not discussed.

Kleber and schulter (2010) presented a framework of IDS for Cloud computing network that could reduce the impact of attacks. The implementation results indicate that the proposed system could resist DoS

attack. Moreover, by comparison, the proposed cooperative IDS system only increases little computation effort compared with pure Snort based IDS that can prevent the system from single point of failure attack.

Patel *et al.* (2013) proposed a model for intrusion detection system which combines the concept of autonomic computing, fuzzy logic, and ontology and risk management.

Narwane and Vaikol (2012) proposed a system to detect intrusions in the cloud computing using Behavior-based approach and knowledge-based approach. If first approach is unable to detect the data, second approach again verifies the data and compare it with the signatures within the database. The proposed system can have very low false positive alarm.

Amirreza and Alireza (2012) introduces a Cloud Intrusion Detection System Services (CIDSS) which is developed based on Cloud Computing and can make up for the deficiency of traditional intrusion detection, and proved to be great scalable. CIDSS can be utilized to overcome the critical challenge of keeping the client secure from cyber-attacks while benefit the features which are presented by Cloud Computing technology.

Zhang *et al*. (2019). Proposed a system to detect intrusion using Deep Generative Neural Network (DGNN) that performs Adversarial learning with Data Augmentation in intrusion detection. With the use of data augmentation in intrusion detection, the detection rate and precision was better. But when there is not enough test data for the systemtotrain with, the system experiences data scarcity and imbalance.

3. RESEARCH METHODOLOGY

The methodology adopted is the Object-Oriented Analysis and Design (OOADM). The object-oriented analysis focuses on the definition of classes and the manner in which they collaborate with one another to effect customer requirements. Unified modeling language (UML) and the Unified Process are predominantly features of object oriented Analysis. The Object-Oriented Analysis and Design (OOADM) is a generic model, based on the object oriented paradigm that provides the designer with the semantics and notation necessary for the development of web based interfaces and its connections with previously existing application logic modules (Dayanand *et al.* 2012).

3.1. SYSTEM DESIGN

3.1.1 The New System

The new system adopted the Zhang model and enhanced it by combining two machine learning techniques (support vector machine and Bayesian network) to aid in the classification of normal data and intrusion data. From figure 1. Data are collected and moved to the Data pre-processing stage, where they are normalized, to reduce the height and volume of the data, and then they are classified into Normal Data (ND) or Intrusion Data (ID) using the two machine learning technique (support vector machine and Bayesian network). Both the Normal Data and Intrusion data are sent to the next stage which is the Data Partition where the Normal Data are now classified as Normal Network Request while the Intrusion data are classified as Network Intrusion. The Normal Network request are sent through the Date management and Data aggregation process to the shadow learning, while the Network Intrusion (Intrusion Data) go through Data Augmentation process of Zhang (Zhang 2019) and finally to the Deep learning Training Model. The above description constitute the Training Phase for the models.

The testing phase include feeding the model with test data, which are then normalize and classified and a result of weather they are intrusion Data (ID) or Normal Data (ND) is made based on the result from the shadow learning process or Deep learning process of the trained model.

3.1.2. MODEL OF THE NEW SYSTEM

TRAINING PHASE

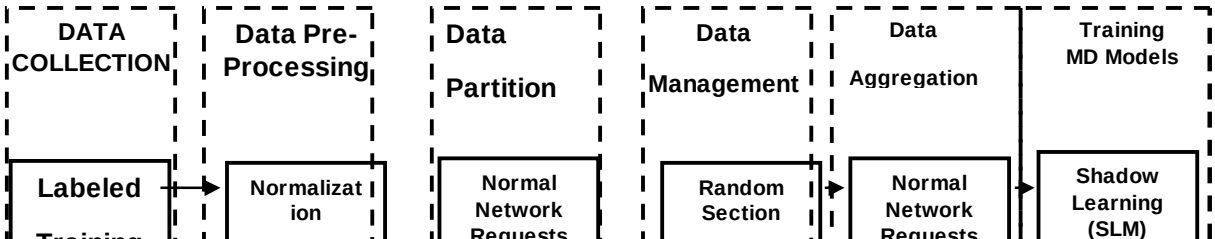

Figure 1. Model View of the New System

3.2. High level Model of the new System

The high level model explains the architecture that would be used for developing the automated system. The high level diagram shown in figure 2 provides an overview of the entire system, identifying the main components that would be developed and their interfaces.

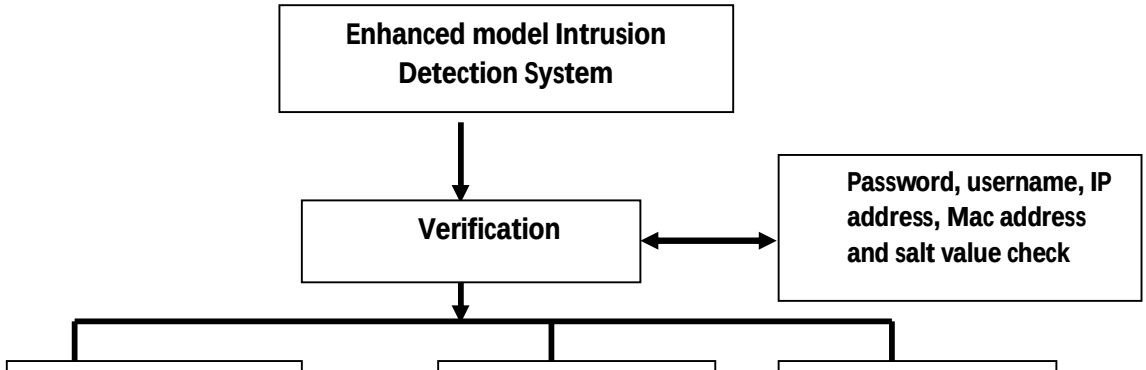

Figure 2. High level Model view of the new system

4. IMPLEMENTATION

4.1. HARDWARE REQUIREMENTS

Hardware requirements for the project designed to run on a standalone computer and/or on server are as follows:

a. Processor- Pentium III or Pentium ® Dual core 2.0GHz or higher
b. Memory- 3.00GB of Random Access Memory (RAM). The physical total size of the software is 2.0MB and the database with samples user details is 1.5KB and is expected to increase when opened for commercial use.
c. A minimum of 1.0GB of free hard disk space.
d. A network Interface Card (NIC) or Wireless Local Area Network (WLAN) for network connectivity.
e. A standard Video Graphic Array (VGA)
f. A monitor for display.
g. Keyboard and mouse for input and pointing

h. CD-ROM drive or external CD-ROM drive

4.2. SOFTWARE REQUIREMENTS

The software requirements for the developed software include:

a. Operating system (OS) – Windows 8 or higher version, Unbutu
b. Apache Server 2.2.17 or higher
c. MySQL5.5.8 or higher
d. PHP 5.3.5 Interpreter or higher
e. Good Web browser software.
f. Reliable access to the internet

4.3. SYSTEM TESTING

In testing, we check both verification (that is if the software compiles with the requirements) and validation (that is if the software has been written correctly and effectively). The software was also tested against its analysis specification.

There were different types of tests adopted at different stages. They include:

i) Unit Testing – testing each class or unit of the software interface.
ii) Integration Testing – testing done during the combination of various class and various modules for compatibility check.
iii) Module/Sub-System Testing – Testing done when on a module before integration.
iv) System Testing – testing after the combination of the various modules or subsystems to produce the required software.
v) Acceptance Testing – In this stage, we invited people who work in an academic environment that uses a private cloud with full internet access to do the acceptance testing.

Two different software testing techniques were adopted as a systematic testing approach they include:

i) White Box Testing – This technique focused on the program control structure which involved close examination of procedure derail. Program statements, internal data structure, loop, logical paths and logical statements were tested. White box testing helped us to test the quality of the construction of the software.

ii) Black Box Testing –This technique tested the quality of the performance of the software and was conducted at the software interface. It tested the functionality of the system.

The aim of the two test technique conducted was to ensure that the software has the following attributes: Completeness, Correctness, Reliability and possibility of maintenance.

5. RESULTS AND DISCUSSION

A test data was obtained with the use of data packet receiver and classification of the intrusions as shown in table 1.The result gotten from the classification of the intrusions was used to test for true positive and true negative as shown in table 2. And from this, a comparison/performance evaluation was done to ascertain if the enhanced intrusion detection model was better and why it was better than the existing one as shown in table 3.

Table 1.: Classification of the intrusions used to test for true positive and true negative

TRUE POSITVE	**FALSE POSITVE**
Reality: An intrusion Attack occurs	**Reality:** No intrusion Attack occurs
Enhanced IDS: Detects an Attack occurs	**Enhanced IDS:** Detects an Attack occurs
Output: Record the attack as TP	**Output:** Record the attack as FP

Table1.Illustrating the way the result table is gotten

ID	frame	byte	Result
1	Frame 1:	208	2.05807365439093
2	Frame 2:	42	1
3	Frame 3:	42	1
4	Frame 4:	208	2.05807365439093
5	Frame 5:	208	2.05807365439093
6	Frame 6:	208	2.05807365439093
7	Frame 7:	208	2.05807365439093
8	Frame 8:	208	2.05807365439093
9	Frame 9:	208	2.05807365439093
10	Frame 10:	208	2.05807365439093
11	Frame 11:	208	2.05807365439093
12	Frame 12:	92	1.31869688385269
13	Frame 13:	92	1.31869688385269

ID	frame	byte	Result
14	Frame 14:	208	2.05807365439093
15	Frame 15:	92	1.31869688385269
16	Frame 16:	80	1.24220963172805
17	Frame 17:	158	1.73937677053824
18	Frame 18:	66	1.15297450424929
19	Frame 19:	66	1.15297450424929
20	Frame 20:	54	1.07648725212465
21	Frame 21:	54	1.07648725212465
22	Frame 22:	72	1.19121813031161

Table 2.Showing the Expected and Actual result

5.1. PERFORMANCE EVALUATION

Number of Packet Frame Analyzed	Number of True Positive	Number of True Negative
12362	1908	10454
6120	954	5166
12240	1908	10332
6006	945	5061
5945	945	5000
5823	943	4880
11829	1888	9941

Table 3. Performance evaluation carried out on the new system

From table 3. It shows that the new enhanced intrusion detection system was more reliable than that of Zhang, as seen in the high rate of true positive and true negative value.

5.2 CONCLUSION

Networks security problems vary widely and can affect different security requirements including authentication, integrity, authorization, and availability. Intruders can cause different

types of attacks on systems in an organization. These attacks need to be detected as soon as possible to prevent further damages to organizations sensitive data which may cause financial loss.

The enhanced model for intrusion detection developed, brought about a new method of detecting intruders by adopting the Zhang model and enhancing with the combination of two machine learning techniques to aid in the classification of normal and intrusion data. By this process improved on the existing system in terms of providing more accurate and more efficient intrusion detection. It also provided worthwhile information about malicious network traffic; helping to identify the source of the incoming probes or attacks; collecting forensic evidence that can be used to identify intruders and alerting security personnel that a network invasion maybe in progress.

REFERENCES

Abraham, A., Jain, R., Thomas, J. & Han, S. Y. (2017). D-SCIDS: Distributed soft computing intrusion detection system. Journal of Network and Computer Applications, January, 30(1), 81-98. Elsevier.

AmirrezaZarrabi, AlirezaZarrabi,(2012): "Internet Intrusion Detection System Service in a Cloud"IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 5, No 2,ISSN (Online): 1694-0814.

Dayanand Ingle1 and Dr. B.B. Meshram (2012): Hybrid Analysis and Design Model for Building Web Information System. IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 4, No 3, July 2012 ISSN (Online): 1694-0814.www.IJCSI.org

Gupta. S, Horrow.S and Sardana.A.(2012):"A Hybrid Intrusion Detection Architecture for Defense against DDoS Attacks in Cloud Environment," Contemporary Computing Communications in Computer and Information Science, Vol. 306, ISBN: 978-3-642-32129-0, pp. 498-499.

Hwang .K, Cai. M, Chen Y., Member. S, and Qin.M (2007): "Hybrid Intrusion Detection with Weighted Signature Generation over Anomalous Internet Episodes", *IEEE Transactions on Dependable and Secure Computing*, 4(1), pp. 1-15.

Irfan Gul, Hussain M.,(2011):"Distributed cloud intrusion detection model", International Journal of Advanced Science and Technology Vol. 34.

Kanubhai et al (2013) worked onAn Architecture of Hybrid Intrusion Detection System.

Kleber, schulter,(2010): "Intrusion Detection for Grid and Cloud computing", IEEE Journal: IT Professional.

Kholidy, Hisham A. Baiardi F.(2012): CIDS: A framework for intrusion detection in cloud systems. Proceedings of 9th IEEE International Conference on Information Technology-New Generations.p. 379–85.

1 MATTHEW,ERNST AND YOUNG (2015) SANS BOSTON: INTRUSION DETECTION FAQ: WHAT IS INTRUSION DETECTION? HTTP://WWW.SANS.ORG/SECURITY-RESOURCES/IDFAQ/WHAT_IS_ID.PHP. RETRIEVED 3RD APRIL 2015.

Narwane S.V., Vaikol S. L.(2012):"Intrusion Detection System in Cloud Computing Environment" International Conference on Advances in Communication and Computing Technologies (ICACACT) Proceedings published by International Journal of Computer Applications (IJCA).

Patel A, Taghavi M, Bakhtiyari K, Celestino J,Junior(2013):. An intrusion detection and prevention system in cloud computing: A systematic review. Journal of Network and Computer Applications. 36(1): 25–41.

Patel A, Taghavi M, Bakhtiyari K, Celestino J,Junior(2013):. An intrusion detection and prevention system in cloud computing: A systematic review. Journal of Network and Computer Applications. 36(1): 25–41.

Roschke Sebastian, Cheng Feng, Christoph Meinel,(2009): “Intrusion Detection in the Cloud", Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing.

Stillerman.M, Morceau.C, Stillman. M.(1999) Intrusion detection for distributed application. Communication of the ACM, 42(7).page 62-69.

Tupakula U, Varadharaja V, Akku N.(2011):Intrusion detection techniques for infrastructure as a service cloud. Proceedings of 9th IEEE International Conference on Dependable, Autonomic and Secure Computing. p. 744–51.

Xin W, Ting-Lei H, Xiao-Yu L. (2010):Research on the Intrusion detection mechanism based on cloud computing. Proceedings of International Conference on Intelligent Computing and Integrated Systems; Guilin.p. 125–8.

Zhang He, Xingrui Yu, Peng Ren, Chumbo Luo, Geyong Min (2019). Deep Adversial learning in intrusion detection. A data Augmentation EhancedFramework.ArXiv: 1901.07949v3 [CS.CR]