THE INTERNATIONAL JOURNAL OF
SCIENCE & TECHNOLEDGE

A Model for Detecting Information Technology Infrastructure Policy Violations in a Cloud Environment

Ruth Oginga
Lecturer, Department of Computer Science and IT, Kabarak University, Kenya
Felix Musau
Senior Lecturer, Department of Computer Science, Riara University, Kenya
Christopher Maghanga
Senior Lecturer, Department of Computer Science and IT, Kabarak University, Kenya

Abstract:
The pervasiveness of the internet and available connectivity solutions brought about by cloud computing has led to unprecedented increase in technologies built based on information technology infrastructures. Most organizations consider the deployment of different types of protection systems to curb the various malicious activities. Organizations offer sophisticated monitoring and reporting capabilities to identify attacks against cloud environment, while stopping multiple classes of attacks before they successful interfere with network activities. Users with ill intentions have increasingly used the cloud as an attack vector due to its ubiquity, scalability and open nature despite the existence of policy violation detection systems necessitating the need to strengthen access policies from time to time. Policy violation detection plays a major role in information security by providing a systematic way of detection and interpreting attacks. Some of the known weaknesses of most detection tools are the generation of false positives or false alerts and inability to perform analysis if traffic is encrypted as well as failure to detect and prevent attacks. This research paper was concerned with the investigation of weaknesses of firewall and Intrusion Detection system (IDS) which are supported by the cloud. The research design for the paper was based on the mixed methods. Experimental results revealed weakness in existing systems specifically IDS and firewall. Unlike the existing systems, new model designed to overcome the shortfall was able to detect both known and unknown attacks and signatures. Moreover, the model was capable of preventing the occurrence of false positives, and terminates suspicious nodes in real time without human intervention. Based on the tests carried out, it was recommended that Policy violation detection model be implemented to guarantee protection. An additional area of application such as migration from one cloud to another is not achievable, at this moment because of the heterogeneous nature of the cloud. This is a potential area for investigation in future.

Keywords: *POVIDE model, policy violation, develop, cloud, detection, weaknesses, attacks*

1. Introduction

The number of business organizations moving towards cloud is increasing very rapidly. The ease of use and the connectivity the cloud provides is highly useful but the risks involved and malicious intrusions are also increasing day by day. Intrusions, malware or security policy violations of curious or malicious users are just but a few. Control assets and information policy are required in order to protect organization assets of the cloud-computing environment. Acceptable use policy is needed to make sure controls and monitoring of services is provided. Acceptable use policy is a set of rules applied by organizational network administrators to restrict the ways in which the network is used and set guidelines as to how it should be used (Fontijn et al., 2015).

Different network administrators use different types of network-based and host-based security software to detect malicious activities in the cloud. The main target of the assailants is to make an attack to the presented resources in the Cloud computing settings (Hameed et al., 2016).

Intrusion is the act of violating the security policy that pertains to an information system. Intrusion detection can be defined as the act of detecting actions that attempt to compromise the confidentiality, integrity or availability of a resource (Adat, & Gupta, 2018).

Various methods can be used to detect intrusions but each one is specific to a specific method. The main goal of an intrusion detection system is to detect the attacks efficiently. Furthermore, it is equally important to detect attacks at the beginning stage in order to reduce their impacts (Schwarz et al., 2017).Most organizations today make use of acceptable use policy to specify the actions prohibited to the users of an organization's IT infrastructure. All users are usually required to adhere to all the policies specified in the acceptable use policy document without exception.

Despite the use of existing detection and prevention systems such as IDS, IPS and Firewall to detect and prevent malicious activities and to analyze data that originates from the host computer, some users circumvent detection and prevention tools to access the cloud (Banerjee et., 2018).The greatest challenge with most of the detection and prevention technologies is the generation of false positives or false alerts.The greatest challenge with most of the detection and prevention technologies is the generation of false positives or false alerts. Furthermore, existing detection tools are unable to perform analysis if traffic is encrypted in real time. For these reasons, there is need to develop Policy violation detection model to solve the aforementioned challenges. Therefore there is need to develop Policy violation detection model to solve the aforementioned challenges. Since the developed model would detect Information Technology infrastructure policy violation in a cloud environment. The paper demonstrates at the weaknesses of IDS, IPS and firewall, then it goes ahead to demonstrate those weaknesses. It also evaluates the model for the performance.

As for the remaining part of this paper, Section 1 is introductions which contain problem statement and objectives. This isfollowed by a detailed description of related work in section 2. Section 3 presents a Methodology, POVIDE model testing presented in section 4, Results andfinally, comparison between new and old detection was discussed and conclusions and future work are provided in section.

2. Related Work

There are several studies that have been conducted previously inregard to policy violation in cloud computing . Most of the paper discussed private cloud computing, others public cloud computing and others hybrid. In this section, therefore, a summary of some of the most prominent efforts in previous research work is provided. To begin with, Ford et al. (2016) developed an adaptive enterprise IDPS free open-source break-in prevention software. Fail2ban, is used to create the data collection agent. Here, all software agents are interconnected to the central behavior analysis database service, collect and record attack meta-information during prior attack attempts. The agents use both real-time and previous data by applying integrating rules from the information analysis method into intrusion prevention policies. However, this proposed system has a high false-positive rate.

According to Xiong (2014), current OpenFlow related works like OpenNetMon and OpenSafe both proposed the network monitoring service in a cloud environment to efficiently collect the traffic usage statistics and detect malicious activities. However, they do not propose a comprehensive solution for a cloud system. These works did not go beyond the stage of detection and are not able to provide further analysis and countermeasures for any attack. The 'detecting and alerting' nature of monitoring solutions demand the human-in-the-loop to inspect the generated security alerts and manually take actions, which cannot respond to attacks in a prompt fashion.

According to Vaquero (2011), there is effectiveness when using services cloud such as IaaS and PaaS in educational fields, especially in teaching advanced Computer Science courses. The Blue Sky cloud framework was presented by Goyal (2014) to implement a cloud that supports scalable and cost-efficient for the E-learning system for basic education in China.

According to Sodhi and Prabhakar (2011), pure autonomous system architecture based on IaaS where control of cluster nodes is fully autonomous was presented. This model uses real-time information from cluster nodes and decentralizes the policy management from the master node to other working nodes. It has several main components namely cloud controller, a gateway for clients into cloud, which determines the suitable node to run VM that satisfies client's needs, Cloud agent an intelligent software component that responds to the queries of the cloud controller regarding the availability of VM configuration for a specific lease duration. It also contain VM foundry, VM image repository interface dedicated to answer queries for particular VM configurations and it creates the one-time-URLs for the VM image. The cloud agent is further based on several components including request handler, VM manager, policy manager, capability manager, and data store (Sodhi, & Prabhakar, 2011).

2.1. Experimental Model on Curbing the Weaknesses of IDS and Firewall on Policy Violation in Cloud

According to Souley and Abubakar (2018), CAPTCHA is a tool commonly used in IPS, to prevent machine intruders (bots) from intruding into a system, however, in this research work, CAPTCHA was used as IDS. The technique to be used is cognizance of the fact that there are software intruders that can read CAPTCHA and attempt to infiltrate it and intrude into the system. CAPTCHAs with weak design pattern and fixed length with varying colors on the text was employed for use in web-based system acting like IPS while in real sense it is an IDS that will attempt to lewd software intruders using machine learning-based attack to successfully read the text-based CAPTCHA and infiltrates the system.

2.2. Demonstrate the Weaknesses of Existing Detection Tools on Policy Violation in the Cloud

According to Lo et al., (2010) proposed and implemented IDS that worked in a supportive way to oppose the DoS and DDoS attacks. It consisted of four components. The first component performed intrusion detection by collecting and analyzing the network packets. The second component immediately drops the packets and checks whether it is correspondence with the block table rules or not, if packets having no autonomic element manager, autonomic coordinator, and correspondence to these rules are forwarded to the alert clustering module which generates alert for the suspicious packet. The third component blocks the suspicious packets and sends alerts to other IDSs. The fourth component collects alerts and makes a decision about the packet. They could protect the system from a single point of failure attack by deploying the above-proposed IDS. However, it cannot detect unknown attacks since it uses signature-based detection techniques to detect intrusions.

2.3. Develop a Model to Detect and Identify Policy Violation in the Real-Time Traffic

According to Gul and Hussain (2011), an efficient model that uses multithreading technique for improving IDS performance within the Cloud computing environment to handle a large number of data packet flows was suggested. The proposed multi-threaded NIDS is based on three modules namely capture module, analysis module and reporting module.

The first one is responsible for capturing data packets and sending them to analysis part which analyzes them efficiently through matching against a pre-defined set of rules and distinguishes the bad packets to generate alerts

3. Methodology

The research design for this study was based on the mixed methods research design. Mixed method research design has the advantages of using multiple ways to explore a research problem. Here the researcher used experimental design and survey of literature for gathering requirement and development of the model. Senior network administrator represented sample group to come up with the different institutions and organization as they represented the universities and organizations in Kenya

3.1. Test on Reliability and Validity

Reliability was tested through a pilot study that was undertaken in different organizations. The reliability of the instruments and the scales to be used in the study was established using Cronbach's Alpha (Cronbach, 1951). It is a widely accepted measure of internal reliability and consistency that works by identifying items in the instrument that have low correlations in order to exclude them from further analysis. From the reliability analysis, it can be observed that the value of Cronbach's alpha is 0.718 which is greater than 0.7 alpha. This means that the scale conforms to internal reliability. This means that the researcher can use the tool for evaluating the expert opinion.

4. Results

It entailed the involvement of selected domain experts to test the POVIDE Model in the policy violation issues and comment on the extent to which they thought the model represented real situation while utilizing the cloud. The specific individuals who tested the model were chief network administrators. In order to have uniform feedback from the chief network administrators, they were required to fill the evaluation form while testing. They were guided on how to maneuver in the model. The model also shows the number of individuals who accessed the model as show in the Figure 1. To determine the relationship between the perception of the ICT experts, the results were fitted using Poisson regression in Log Linear Model with response variable as the count and the distribution being Poisson while the link function was logarithm; $\eta = \log(\mu)$. The outputs are presented on Table 1 and Table 1.2(b)

Criterion	DF	VALUE	VALUE/DF
Deviance	8	21.8031	2.7254
Scale Deviance	8	21.8031	2.7254
Pearson Chi-square	8	19.994	2.4993
Scaled Pearson X^2	8	19.994	2.4993
Log likelihood		1344.49	

Table 1: Criteria for Assessing Goodness of Fit

Standard				Wald 95 confidence			
Parameter	DF	Estimate	Error	Lower limit	Upper limit	CHISQ	P>CHISQ
Intercept	1	3.8067	0.0563	3.9171	3.9171	4564.57	<0.0001
Perception	1	0.1446	0.0979	-0.0473	0.3365	2.18	0.14
Perception	0	0	0	0	0		
Scale	0	1	0	1	1		

Table 2: Analysis of Parameter Estimates
Source: Author's, 2019

The analysis indicates that the perception of the expert does not depended on the ranks the expert give on the model. This shows that there is independent association between the perception and the rank of the model by the expert. We therefore accept their assessment that where they agreed that the model detects violation of the computer.

4.1. Policy Violation Detection Model

The user can access the POVIDE Model using different technologies such as laptops and phones. The user can access different cloud services depending on cloud service providers. The cloud is divided into layers. The upper layer is Software as a Service (SaaS), which is the one visible to the final user and involves applications. The next layer is Platform as a Service (PaaS) and it matters to software developers. It is composed of the operating systems, application-programming interfaces (API), documentation, and basic services. Infrastructure as a Service (IaaS) refers to the usage of available resources on the cloud: memory, processors, storage and finally business process as a service (BPaaS) as the delivery of business process outsourcing (BPO) services that are sourced from the cloud and constructed for multitenancy. As a cloud service, the BPaaS model is accessed via Internet-based technologies. A cloud management platform is a suite of integrated software tools that an enterprise can use to monitor and control cloud computing resources. While an organization can use a cloud, management platform exclusively for private or public cloud deployment, these toolsset commonly target hybrid and multi-cloud models to help centralize control of various cloud-based infrastructures. Then there is a policy violation detection model that is used to detect any violation on the cloud see Figure 1.

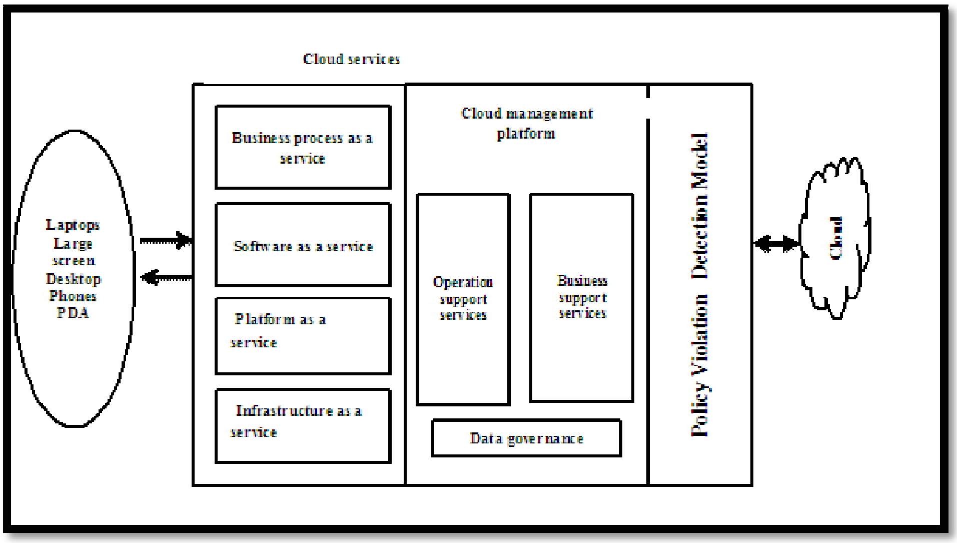

Cloud services
Laptops
Large
screen
Desktop
Phones
PDA
Business process as a service
Software as a service
Platform as a service
Infrastructure as a service
Cloud management platform
Operation support services
Business support services
Data governance
Policy Violation Detection Model
Cloud

Figure 1: Policy Violation Detection Model

4.2. Demonstration of the Existing Weaknesses of the IDS/IPS and Firewall

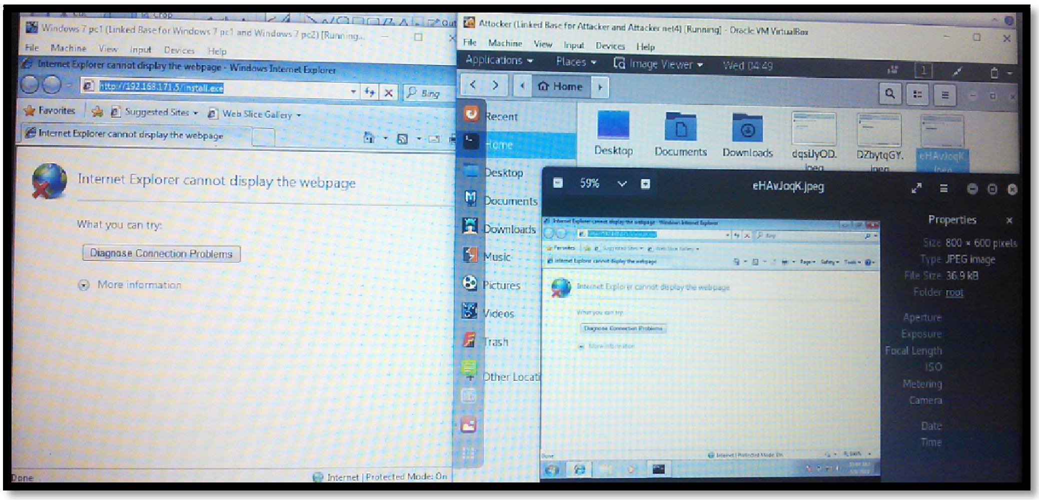
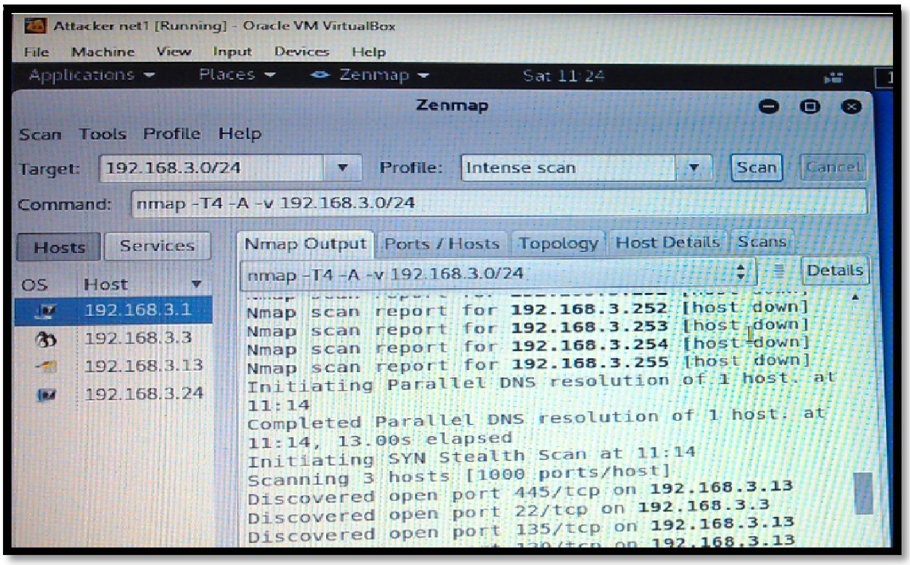

Windows 7 pc1 (Linked Base for Windows 7 pc1 and Windows 7 pc2) [Running...
File Machine View Input Devices Help
Internet Explorer cannot display the webpage - Windows Internet Explorer
http://192.168.171.5/install.exe
Bing
Favorites
Suggested Sites
Web Slice Gallery
Internet Explorer cannot display the webpage
What you can try:
Diagnose Connection Problems
More information
Done
Internet | Protected Mode: On
Attacker (Linked Base for Attacker and Attacker net4) [Running] - Oracle VM VirtualBox
Applications
Places
Image Viewer
Wed 04:49
Home
Recent
Desktop
Documents
Downloads
Music
Pictures
Videos
Trash
Other Locations
dqsUyOD.jpeg
DZbytqGY.jpeg
eHAvJoqK.jpeg
59%
Properties
Size 800 × 600 pixels
Type JPEG image
File Size 36.9 kB
Folder root
Aperture
Exposure
Focal Length
ISO
Metering
Camera
Date
Time

Figure 2: Virtual Machine versus the Attacker's Machine

Figure 2 shows an attacked machine on the left and the right side shows that an attacker can see everything the being done in the virtual machine. The attacker has gained access and can do anything with the virtual machine. Figure 2 shows files downloaded by an attacker from the virtual machines where they are stored. All the information acquired the virtual machine are stored in the home directory and can be accessed by the attacker. This shows the attacker has attacked the virtual machine.

4.2.1. Scanning by the Internal Attacker

Attacker net1 is a virtual machine on the network. Here the internal attacker can scan the network as shown in Figure 3. The exploiting tool used for scanning. Intense scanning is a compressive scanning of the network. It scans all TCP ports. While a scan is running and after it completes, the output of the nmap command is shown on the screen.


Attacker net1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Zenmap Sat 11:24
Zenmap
Scan Tools Profile Help
Scan
Cancel
Hosts
Services
Nmap Output
Ports / Hosts
Topology
Host Details
Scans
Details
OS
Host

Target: 192.168.3.0/24 Profile: Intense scan

Command: nmap -T4 -A -v 192.168.3.0/24

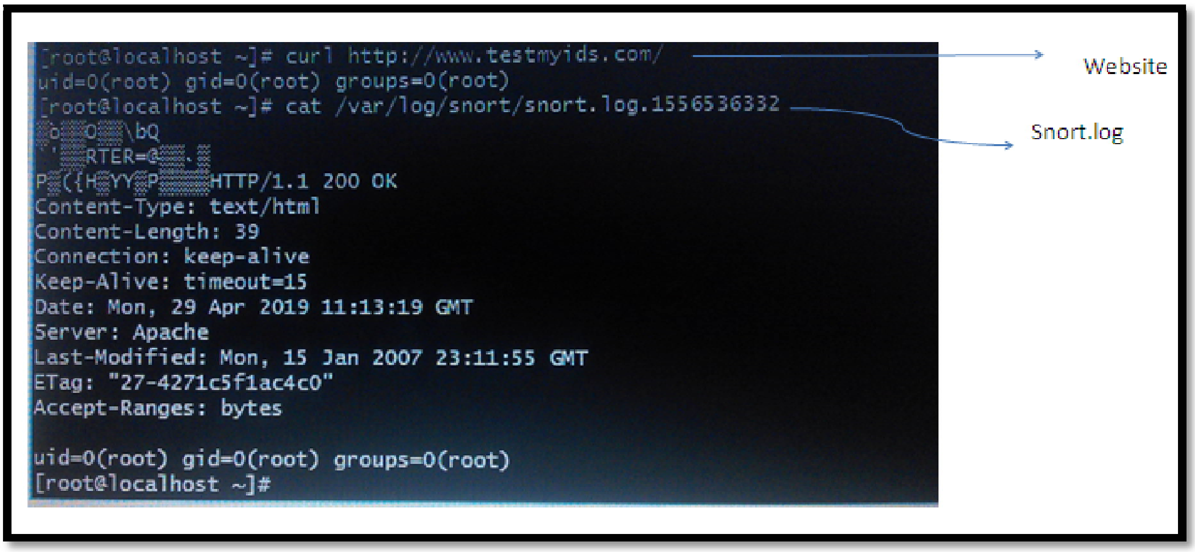
```
192.168.3.1
192.168.3.3
192.168.3.13
192.168.3.24
```

```
nmap -T4 -A -v 192.168.3.0/24
Nmap scan report for 192.168.3.252 [host down]
Nmap scan report for 192.168.3.253 [host down]
Nmap scan report for 192.168.3.254 [host down]
Nmap scan report for 192.168.3.255 [host down]
Initiating Parallel DNS resolution of 1 host. at
11:14
Completed Parallel DNS resolution of 1 host. at
11:14, 13.00s elapsed
Initiating SYN Stealth Scan at 11:14
Scanning 3 hosts [1000 ports/host]
Discovered open port 445/tcp on 192.168.3.13
Discovered open port 22/tcp on 192.168.3.3
Discovered open port 135/tcp on 192.168.3.13
```

Figure 3: Scanning by Internal Network

The nmap-T4-A-V 192.168.3.0/24command was used to scan the entire network server for any opened ports by the internal attacker. Then the nmap command was used to scan and report the IP addresses. The results were as expected, as shown in Figure above. The figure also showed that 192.168.3/24was used to view the content of the whole network server and scanning details. After scanning the network four host IP addresses were found to be opened, namely 192.168.3.1, 192.168.3.3, 192.168.3.13 and 192.168.3.24. Next, NMAP output was used to view the metadata and display the information.

There were four hosts' machines with opened ports such 445, 22, and 135 which was on the network server. Nmap scans do not use the pivots you have set up. This means that the firewall could not detect threats from the internal attacker. This test was compared to that of Keshri et al. (2016) where they presented a Denial of Service (DoS) prevention technique using a firewall and based on data mining techniques, which comprises data selection, data preprocessing, transformation, and model selection and evaluation. However, the technique could not detect internal attacks. This weakness could be solved by making sure that all open ports are blocked and denied access. This would allow the internal attacker would not be able to scan the network. This is the example of the false alert, while site accessed is actually a website. The results as shown in Figure 4


```
[root@localhost ~]# curl http://www.testmyids.com/
uid=0(root) gid=0(root) groups=0(root)
[root@localhost ~]# cat /var/log/snort/snort.log.1556536332
▒o▒▒O▒▒\bQ
 '▒▒RTER=@▒▒.▒
P▒({H▒YY▒P▒▒▒▒HTTP/1.1 200 OK
Content-Type: text/html
Content-Length: 39
Connection: keep-alive
Keep-Alive: timeout=15
Date: Mon, 29 Apr 2019 11:13:19 GMT
Server: Apache
Last-Modified: Mon, 15 Jan 2007 23:11:55 GMT
ETag: "27-4271c5f1ac4c0"
Accept-Ranges: bytes

uid=0(root) gid=0(root) groups=0(root)
[root@localhost ~]#
```

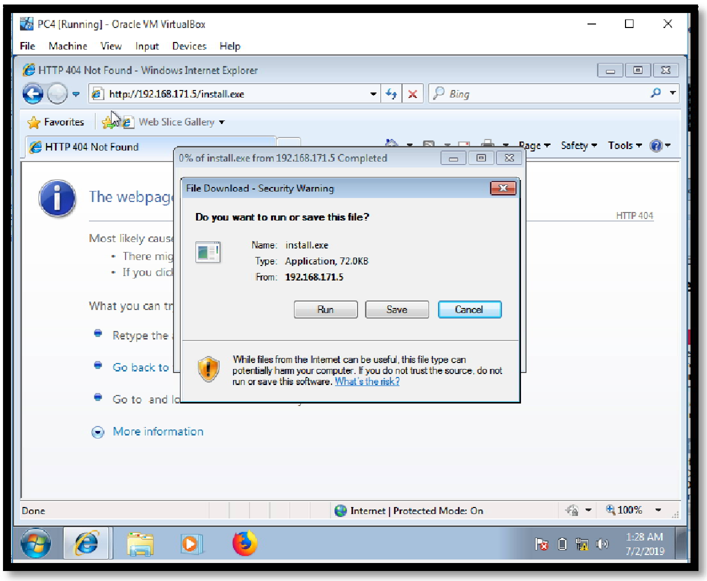
Figure 4: False Alert

curlhttp://www.testmyids.com/ this a website the user's pc is accessing. The website was found to contain data such uid=0(root) gid=0(root) groups=0(root).

cat/var/log/snort/snort.log.1556536332 commands read the alert log from snort and gather the list of critical ports to give a report of the file. It was found out that snort gave a report of the live attack. The report also shows the content type, length of the file, date and time accessed. It further gives the server the file is received from and the last time the file was modified. The result was as expected, as shown in Figure 4. This result was compared to Ford et al. (2016) who developed an adaptive enterprise IDS. Free open-source break-in prevention software and Fail2ban used to create the data collection agent. The agents used both real-time and previous data by applying integrated rules from the information analysis method into intrusion prevention policies. However, this proposed system had a high false positive rate. This weakness of false positive can be curbed by developing a strong technique that would only detect an alert when it happened.

4.3. Povide Model Results

When using POVIDE model, the attack was detected and blocked the attack. The Figure 5shows an example of threat sent by an attacker and accessed by the virtual machine. The Figure 6 is the POVIDE Model shows the results after detecting and blocking the attack on the VM. It shows the name of the threat, where it's coming from and which IP address affected. It shows that the POVIDE Model would detect and block the attacks in real time thus eliminating false negative. It monitors the events on the network, inspects the data and collects evidence of intrusive behaviors. Whenever it detects suspicious or malicious attempts, it signals an administrator instantly for a reaction.


PC4 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
HTTP 404 Not Found - Windows Internet Explorer
http://192.168.171.5/install.exe
Bing
Favorites
Web Slice Gallery
HTTP 404 Not Found
Page Safety Tools
0% of install.exe from 192.168.171.5 Completed
File Download - Security Warning
The webpag
HTTP 404
Most likely caus
There mig
If you dic
What you can tr
Retype the
Go back to
Go to and lo
More information
Run
Save
Cancel
Done
Internet | Protected Mode: On
100%
1:28 AM
7/2/2019

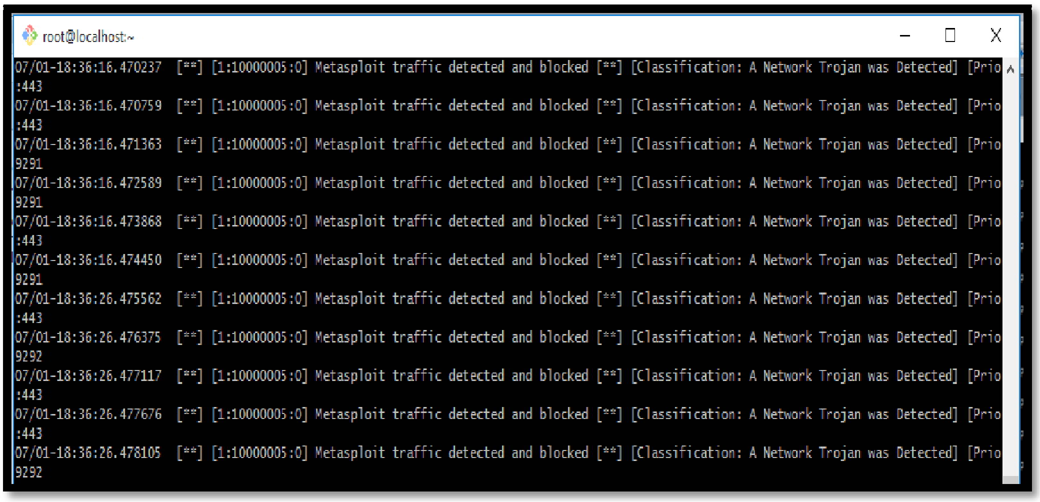
Do you want to run or save this file?

Name: install.exe
Type: Application, 72.0KB
From: 192.168.171.5

While files from the Internet can be useful, this file type can potentially harm your computer. If you do not trust the source, do not run or save this software. What's the risk?

Figure 5: A Threat Website

4.3.1. Threat Detected and Blocked



```
07/01-18:36:16.470237  [**] [1:10000005:0] Metasploit traffic detected and blocked [**] [Classification: A Network Trojan was Detected] [Prio
:443
07/01-18:36:16.470759  [**] [1:10000005:0] Metasploit traffic detected and blocked [**] [Classification: A Network Trojan was Detected] [Prio
:443
07/01-18:36:16.471363  [**] [1:10000005:0] Metasploit traffic detected and blocked [**] [Classification: A Network Trojan was Detected] [Prio
9291
07/01-18:36:16.472589  [**] [1:10000005:0] Metasploit traffic detected and blocked [**] [Classification: A Network Trojan was Detected] [Prio
9291
07/01-18:36:16.473868  [**] [1:10000005:0] Metasploit traffic detected and blocked [**] [Classification: A Network Trojan was Detected] [Prio
:443
07/01-18:36:16.474450  [**] [1:10000005:0] Metasploit traffic detected and blocked [**] [Classification: A Network Trojan was Detected] [Prio
9291
07/01-18:36:26.475562  [**] [1:10000005:0] Metasploit traffic detected and blocked [**] [Classification: A Network Trojan was Detected] [Prio
:443
07/01-18:36:26.476375  [**] [1:10000005:0] Metasploit traffic detected and blocked [**] [Classification: A Network Trojan was Detected] [Prio
9292
07/01-18:36:26.477117  [**] [1:10000005:0] Metasploit traffic detected and blocked [**] [Classification: A Network Trojan was Detected] [Prio
:443
07/01-18:36:26.477676  [**] [1:10000005:0] Metasploit traffic detected and blocked [**] [Classification: A Network Trojan was Detected] [Prio
:443
07/01-18:36:26.478105  [**] [1:10000005:0] Metasploit traffic detected and blocked [**] [Classification: A Network Trojan was Detected] [Prio
9292
```

Figure 6: False Negative

This is the results of the external attacker trying to attack a user's Virtual machine on the POVIDE model. Meterpreter is the exploiting tool. The file being accessed has been forwarded to the POVIDE model using root/dqsiJyoD.jpeg for analysis. Idle time command is to show the time the machine has been idle. Shell is a command to show the kind of machine that is being attacked. The results are as shown in the Figure 6. The POVIDE Model is able to detect an attack as it happens if the attacker bay passes the blocked ports. The model would also check on the different processes created. This would help in detecting the false negative. This experiment showed that any action performed inside the VM is not recorded in the log file. Examples include restarting a VM, formatting a disk, and copying a file. On the other hand, actions performed via POVIDE Model or CLI are recorded.

	IDS	**Firewall**	**IPS**	Povide
General features	Detects attacks but cannot prevent the attacks	Cannot react to a network nor initiate effective counter measures	Prevent attacks but cannot detect attacks	Detects and prevents attacks in real time
Anomaly Detection	Detects known attacks	Detects known attacks	Cannot detect known attacks but prevent known attacks	Detects both known and unknown attacks
Signature Detection	Cannot detect unknown signatures	Cannot detect unknown signatures	Cannot detect unknown signature	Detects both known and unknown signature
Human intervention is required to investigate the attacks once its detected	yes	yes	yes	No
Detects Internal attacks	Can Detect	Cannot Detect	Cannot Detect	Can Detect
False Positive	Produces high false positive	Produces high false positive	Produces high false positive	Prevents false positive
Encrypted packets	Encrypted packets are not processed	Encrypted packets are not processed	Encrypted packets are not processed	Detects and blocks both encrypted and decrypted packets

Table 3: Comparison of Existing Detection Systems Verses POVIDE Model

5. Conclusion

Increasing cost of cybercrime and the growing adoption of Cloud by organizations has demonstrated that there is need for policy violation detection model in the cloud environment.

In conclusion, this research demonstrates the use of this model in the cloud to curb policy violation. Nevertheless, the model is meant to detect hosts violating policies by terminating suspicious nodes in real time.

6. Policy Recommendations

Policies should be strictly implemented in clouds. Organizational and governing bodies should visit clouds' staff and student's infrastructure on regular bases to evaluate the efficiency of the security precautions adopted by the suppliers.

The government should allow cloud service providers and institutions to incorporate and punish users who violate policies as a breach of contract. The government must have national cloud policy, laws and standardized SLA to prevent cloud clients from exploitation since CSP has an upper hand and secretion in implementing the SLA.

Future work include, an additional area of application such as migration from one cloud to another is not achievable, at this moment because of the heterogeneous nature of the cloud. This is a potential area for investigation.It is suggested that a further extension of the prototype should be developed to locally cache big data at remote and to enable more efficient threat detection.

7. References

i. Adat, V., & Gupta, B. B. (2018). Security in Internet of Things: issues, challenges, taxonomy, and architecture. *Telecommunication Systems*, *67*(3), 423-441.

ii. Chowdhary, M., Suri, S., & Bhutani, M. (2014). Comparative study of the Intrusion Detection System. *International Journal of Computer Sciences and Engineering*, *2*(4), pp. 197-200.

iii. Fontijn, W. F. J., Talstra, J. C., Newton, P. S., & Holtman, K. J. G. (2015). *U.S. Patent No. 9,202,045*. Washington, DC: U.S. Patent and Trademark Office.

iv. Ford, M., Mallery, C., Palmasani, F., Rabb, M., Turner, R., Soles, L., & Snider, D. (2016). A Process to Transfer Fail2ban Data to an Adaptive Enterprise Intrusion Detection and Prevention System. *Proceedings of the 2016 IEEE SoutheastCon*, March 31-April 3, 2016, Norfolk, VA.

v. Goyal, S. (2014). Public vs private vs hybrid vs community-cloud computing: A critical review. *International Journal of Computer Network and Information Security*, *6*(3), 20.

vi. Gul, I., & Hussain, M. (2011). Distributed cloud intrusion detection model. *International Journal of Advanced Science and Technology*, *34*(38), 135.

vii. Hameed, A., Khoshkbarforoushha, A., Ranjan, R., Jayaraman, P. P., Kolodziej, J., Balaji, P., & Khan, S. U. (2016). A survey and taxonomy on energy-efficient resource allocation techniques for cloud computing systems. *Computing*, *98*(7), pp. 751-774.

viii. Hock, F., & Kortis, P. (2015). Commercial and open-source based Intrusion Detection System and Intrusion Prevention System (IDS/IPS) design for IP networks. In: *Emerging eLearning Technologies and Applications (ICETA), 2015 13th International Conference on*, pp. 1-4.

ix. Jabez, J., & Muthukumar, B. (2015). An Intrusion Detection System (IDS): Anomaly detection using outlier detection approach. *Procedia Computer Science*, *48*, pp. 338-346.

x. Keshri, A., Singh, S., Agarwal, M., & Nandiy S. (2016). DoS attack prevention using IDS and data mining. *International Conference on Accessibility to Digital World (ICADW)* (pp. 87-92) Guwahat.

xi. Lo, C. C., Huang, C. C., & Ku, J. (2010, September). A cooperative intrusion detection system framework for cloud computing networks. In *2010 39th International Conference on Parallel Processing Workshops* (pp. 280-284). IEEE.

xii. Schwarz, M., Weiser, S., Gruss, D., Maurice, C., & Mangard, S. (2017, July). Malware guard extension: Using SGX to conceal cache attacks. In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment* (pp. 3-24). Springer, Cham.

xiii. Sodhi, B., & Prabhakar T. (2011).A cloud architecture using smart nodes. In: Proceedings of the *2011 IEEE Asia-Pacific Services Computing Conference* (APSCC) pp. 116–123, Jeju Island, Korea.

xiv. Souley, B., & Abubakar, H. (2018). A captcha–based intrusion detection model. *Int. J. Softw. Eng. Appl*, *9*(1), pp. 29-40.

xv. Vaquero, L., (2011). EduCloud: PaaS versus IaaS cloud usage for an advanced computer science course*. IEEE Transactions on Education*, *54* (4), pp.590–598.

xvi. Xiong, Z. (2014). *An SDN-based IPS Development Framework in Cloud Networking Environment* (Doctoral dissertation, Arizona State University).

xvii. Banerjee, M., Lee, J., & Choo, K. K. R. (2018). A blockchain future for internet of things security: A position paper. *Digital Communications and Networks*, *4*(3), 149-160.

xviii. Chang, V., Ramachandran, M., Yao, Y., Kuo, Y. H., & Li, C. S. (2016). A resiliency framework for an enterprise cloud. *International Journal of Information Management*, *36*(1), 155-166.