ASIAN JOURNAL OF INFORMATION TECHNOLOGY

MEDWELL PUBLICATIONS SINCE 2000

Dynamic Correlation based Graded Intrusion Detection Architecture for Cloud

[1]K. Umamaheswari and [2]S. Sujatha
[1]Development Centre, Bharathiar University, Coimbatore, Tamil Nadu, India
[2]Department of Computer Science, Bharathi Womens College, Chennai, Tamil Nadu, India

Key words: Chakravyuha or padmavyuha, Intrusion Detection and Prevention Systems (IDPS), multi-tier defense framework, SVM-SGD, system call analysis, VLAN

Corresponding Author:
K. Umamaheswari
Development Centre, Bharathiar University, Coimbatore, Tamil Nadu, India

Page No.: 171-180
Volume: 19, Issue 9, 2020
ISSN: 1682-3915
Asian Journal of Information Technology
Copy Right: Medwell Publications

Abstract: Data security and privacy are perennial concerns related to cloud migration whether it is about applications, business or customers. The multi-tenant environment is vulnerable to several types of attacks that hackers aiming towards the sensitive data in the broad access area of cloud. Intrusion Detection and Protection Systems (IDPS) are a significant part of the security framework from the beginning of cloud usage. Since, the security framework itself being the primary target of attackers an unbreakable strategy is needed for its protection. In this study, a novel security architecture for cloud environment designed with IDPS components as a graded multitier defense framework. The proposed model is a defense formation of collaborative IDPS components with dynamically revolving alert data placed in multiple tiers of Virtual Local Area Networks (VLANs). Even if many techniques existing for securing the cloud with IDPS, the alert generation delays prevalent due to the static correlation and aggregation. The novel security architecture proposed with two contributions for impregnable protection, one is to reduce alert generation delay by dynamic correlation and the second is to support the supervised learning of malware detection through, system call analysis. The defense formation facilitates malware detection with linear Support Vector Machine-Stochastic Gradient Descent (SVM-SGD) statistical algorithm. The proposed model of Dynamic Correlated, Graded IDPS (DCGIDPS) for cloud requires little computational effort to counter the distributed, coordinated attacks efficiently.

INTRODUCTION

It is becoming a scarce sight in the IT world for Enterprise owned data centers as businesses move to outsource their infrastructure requirements to the cloud provider communities. There is a fundamental shift in the security boundary for enterprisis sensitive data. Hence, there is an increased need for a ubiquitous security approach.

Cloud computing: The great feature of cloud computing is that users can be from anywhere to gain programs, storage and development platforms through the Internet by services offered by Cloud Providers with any of the devices such as PCs, smart phones, laptops or PDAs. The ultimate result is cost savings, availability and scalability[1]. However, the attack surface is increased because of the multi-tenant environment where Cloud users have their sensitive data and applications. There is always a search for better security tool in the world of network security.

Intrusion Detection System (IDS) is one of such tools for alerting any sign of intrusion activities at the Virtual Machine level of virtualized cloud[2]. Intrusion Detection and Prevention Systems (IDPS) includes all protective actions that identification of possible incidents, analyzing log information of such incidents how to block them in the beginning itself and generate reports for the concern of security personnel[3].

IDPS management: The use of IDPS is a necessary addition to security infrastructure. However, it is very much important to secure IDPS components because IDPSs are the primary target of attackers who try to prevent the IDPSs functioning properly to detect attacks or to access the sensitive data on IDPSs like host configuration and known vulnerabilities[4]. The components in IDPSs can be sensors or agents, management and database servers, user and administrator consoles for interaction and management networks. It is highly required to protect software-based IDPS components such that their operating systems and applications are kept fully up-to-date. Some of the protective actions are to create separate accounts for all IDPS users and administrators, not to allow access to all users for IDPS components and ensuring encrypted communications or pass data over a physically or logically separated network.

Virtual Local Area Networks (VLAN): VLANs pave way for logically segregating network traffic on all management communications of the IDPS components. Using of VLAN makes it possible to monitor and control server over a secure network also it can restrict only administration personnel to access the IDPS components. A flexible, reliable and secure networking environment can be obtained from a good VLAN configuration. VLANs are facing numerous attacks based on misconfigurations. VLANs used to segment a network into a collection of isolated networks within the data center. Each of the networks can act as a separate broadcast domain for a set of IDPS components. The proper configuration of VLAN segmentation can severely hinder access to system attack surfaces. Here, only authorized users can see the servers and other devices necessary to perform their management or control tasks. Hence, it is necessary to have a model that configures VLANs for IDPS management components with proper access control settings that can be an impregnable security strategy.

State-of-art in cloud security: While applying IDS/IPS for cloud security, a variety of traditional techniques is available such as signature based detection, anomaly detection, Artificial Intelligence (AI) based detection etc.

Signature-based intrusion detection can detect known attacks only. Several approaches[5-8] use signature based intrusion detection system for detection of intrusion on VMs (front end of cloud environment). Anomaly or behavioral detection alerts anomalous events by comparing with normal behavior[9]. This approach is efficient in the sense that it lowers false alarms for both known and new attacks. This technique can be used for cloud to detect unknown attacks at different levels[10-15]. But a large number of events in the cloud makes it tough to monitor or control using anomaly-based detection. There are many soft computing techniques such as Artificial Neural Network (ANN), Fuzzy logic, Association rule mining, Support Vector Machine (SVM), Genetic Algorithm (GA), etc., available to improve detection accuracy and efficiency of signature based IDS or anomaly detection based IDS[16].

Hybrid techniques combine the advantages of more than one technique. Man and Huh[4] proposed a technique of arranging the IDPS components in a hierarchical manner for handling large scale coordinate attacks. The setup was a collaboration of IDPS components located in various cloud providers networks. Correlation performed only in higher level layers to ensure the clarity of an incident. This incurs a delay in alert generation at higher level components. In ultra- secure-network-architecture, the IDPS components arranged in various tiers separated in distinct DMZs. This model is vulnerable for some incidents aiming at the sensitive data on IDPS components.

Vieira *et al.*[17] presented a Hybrid Intrusion Detection System for Cloud that can detect only selective kind of attacks. Hence, the system cannot be deployed in a real-time distributed environment. Tupakula *et al.*[18] Hybrid Intrusion Detection System cannot handle large-scale, dynamic, multithread and data processing environment. The system has been proposed for Infrastructure as a Service Cloud, hence the synchronization characteristics are not applicable to the system. Kholidy *et al.*[19] framework does not detect the intrusions in a faster manner; the system can handle large scale, dynamic data only partially. On comparing the existing systems[20], it is required to design one dynamic, scalable and a self-adaptive defense framework. Damien Riquet *et al*.[21] discussed the impact of such kind of large-scale attacks on cloud security. Once an alert generated, it is better to process attack data based on system call analysis for further malware detection[22]. Linear Support Vector Machine (SVM) based Stochastic Gradient Descent (SGD) algorithm suitably assists supervised learning of unknown malware detection[23]. The proposed system is a hierarchical defense framework with protection measures against the vulnerabilities present in existing systems.

MATERIALS AND METHODS

Proposed system: The proposed system is a labyrinthine maze of multitier framework organized as a concentric circle of six tiers with each and every tier can be formed by a different set of IDPS components.

Inspirations: The defensive framework for positioning management IDPS components in VLAN is based on 'Padmavyuha'. The Padmavyuha or Chakravyuha is a popularly narrated Military formation in the Indian epic Mahabharata. The Chakravyuha or Padmavyuha is a multi-tier defensive formation that appears like a blooming lotus (Padma) or disc (chakra) when viewed from above. The setup formed as a labyrinth maze where the warriors at each interleaving position would be in an increasingly tough position to fight as shown in Fig. 1.

In Mahabharata, the military formation was used in the battle of Kurukshetra by Dronacharya, who became commander-in-chief of the Kaurava army after the fall of Bhishma Pitamahar. The Chakravyuha was such a special formation that only a few exclusive warriors namely Arjun, Abhimanyu, Krishna and Pradyumna knew how to crack and penetrate.

Figure 1 is the most accurate depiction as it consists of multi tiers and is closer to the depiction of Chakravyuha in the ancient rock carvings and ancient Indian temple structures as well (can find it in Belur of Hassan district in Karnataka) (Fig. 2). Logically, a Chakravyuha should be a multi-layered circular labyrinthine maze where each of the layers are rotating in same or opposite direction in which weak and strong warriors are strategically placed and each of the layers are presented with possible openings which are closely guarded by one of the main highly ranked warriors and his personal troops. The rotating nature of chakravyuha, gave it a unique benefit as the warriors that made the chakravyuha confronted any particular opponent briefly and each people attacked/defended in turn as the formation kept rotating. This special feature effectively nullified the plans from the opponents which they might have devised against any particular warrior within Chakravyuha and thus, confused them off their strategies.

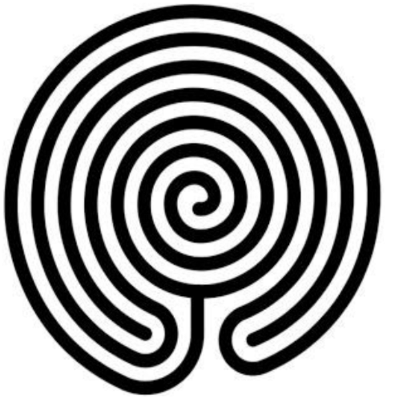

Fig. 1: Padmavyuha or Chakravyuha formation as a labyrinth

Fig. 2: Intricate rock carvings show, Abhimanyu entering the Chakravyuha

This kind of multi-tier defensive formation can be the base for setting VLAN configuration of management IDPS components in the cloud as it never allows any intrusions inside. Even if the intrusions happen at any outer tier it could be caught and blocked at the inner tiers without giving any more time for unwanted entries.

Key considerations: Often it can be found that key innovative techniques in research start their journey from defense. Table 1 shows how the existing issues in applying Intrusion Detection System for Cloud can be solved with Chakra vyuha formation.

Framework of proposed architectural design: Increased security is obtained by moving the sensitive attack data from one component to other within every tier that the sensitive data on attacks readily available for

Table 1: How Chakravyuha fits in IDPS architecture?

Threat issues	Existing IDPS system	Chakravyuha framework
Alert data remains idle in a node until a time limit reaches	Almost the hackers' activity spawned to the entire unit since no action taken immediately	Alert data revolving dynamically for immediate correlation and for further remedial action
Any particular node attacked for the sensitive data in it	A lot of security measures and encryption needed for the particular node	As alert data moving node to node, any particular node will not be attacked for its sensitive data
Alert database can be attacked easily without any extra protection	If alert database got compromised, entire IDS activity will get tampered at once	Alert database placed at innermost tier for increased security
Unknown threats need to be trained manually	No specific measures for improving supervised/ unsupervised learning of malware sources	Supervised learning of malware facilitated by system call analysis of alert generating nodes

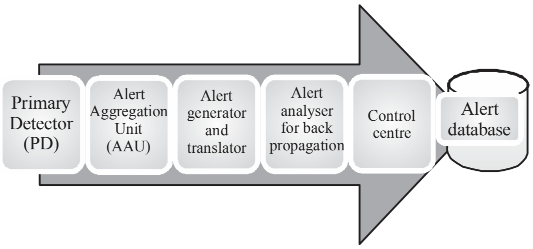
Primary Detector (PD)
Alert Aggregation Unit (AAU)
Alert generator and translator
Alert analyser for back propagation
Control centre
Alert database

Fig. 3: Alert data movement in multi tiers

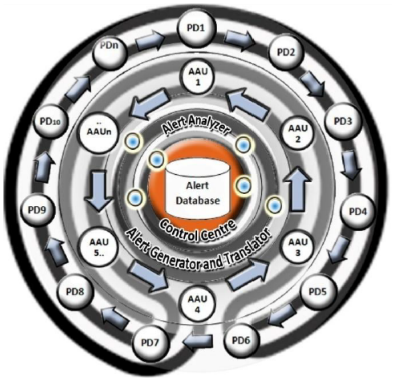

Fig. 4: Multi-tier defensive formation for IDPS management VLAN in cloud

correlation as depicted in Fig. 3 and 4, here, one time server used to synchronize the data movement among components.

Dynamic correlation based graded intrusion detection and prevention system (DCG-IDPS): The outer tier of DCG-IDPS is formed by all the sensors or the agents defined as host level (HIDS) or network level (NIDS) that primarily recognize any malicious event called Primary Detectors (PDs). These components can be controlled by various cloud providers (or a single one) and also interact with the external network. The PDs collect and analyze data about network traffic, memory, file systems, logs, etc. to find potential intrusions in the monitored set of hosts. The key benefit is to reduce alert generation delay by starting correlation of raw alerts in the Primary Detector (PD) level itself with the appropriate alert threshold for each tier and alert data exchanged in real time.

The second tier is formed by IDPS components that aggregate raw alerts based on priority called Alert Aggregation Unit (AAU). AAU collects raw alerts from a set of PDs for reducing the number of false positives and for generating higher level alert reports about large-scale coordinated or multi-step attacks. In AAU, next level of the threshold used for alert aggregation.

The third tier of the formation is the combination of Alert Generator and Translator. The Alert Generator configures all the PDs under corresponding AAU, receives user's data for authentication against blacklisted attackers. Here one local database is maintained for storing configuration and alert data. Alert Translator component translates aggregated alerts into the common format known as IDMEF (Intrusion Detection Message Exchange Format). The translation performed before extracting necessary data and stored in a local database. In the fourth tier, Alert Analyzer component positioned to perform Virtual Machine Introspection (VMI) based on system calls. The VMI process helps the IDS components installed in a privileged domain to monitor the memory state of all Virtual Machines residing on the same physical machine. Furthermore, requests from virtual hosts for I/O devices are also processed by Virtual Machine Monitor and the component does all back propagation activities for blocking the invalid users.

The fifth tier of the defense formation, Control Centre is the management component for information exchange. The control centre finally, acts based on users commands from the console. The final reaction depends on whether an event is truly malicious or not. It correspondingly updates black lists and user configurations in the global database and the data back propagated to the local database whenever necessary. It notifies the users and cloud providers for such kind of cautious events through mails or console messages. The Control Centre handles the cases of other configuration activities such as Virtual Machine migration or removal management as illustrated in Fig. 5.

Distributed Denial of Service (DDoS). It will be like insignificant alert but the severity can be found only after correlation. In DDoS, IDS needs to correlate alerts from multiple attack sources to a single destination but in the case of large-scale stealth scan or worm attack, there will be a correlation of single attack source responsible for numerous alerts to various destinations.

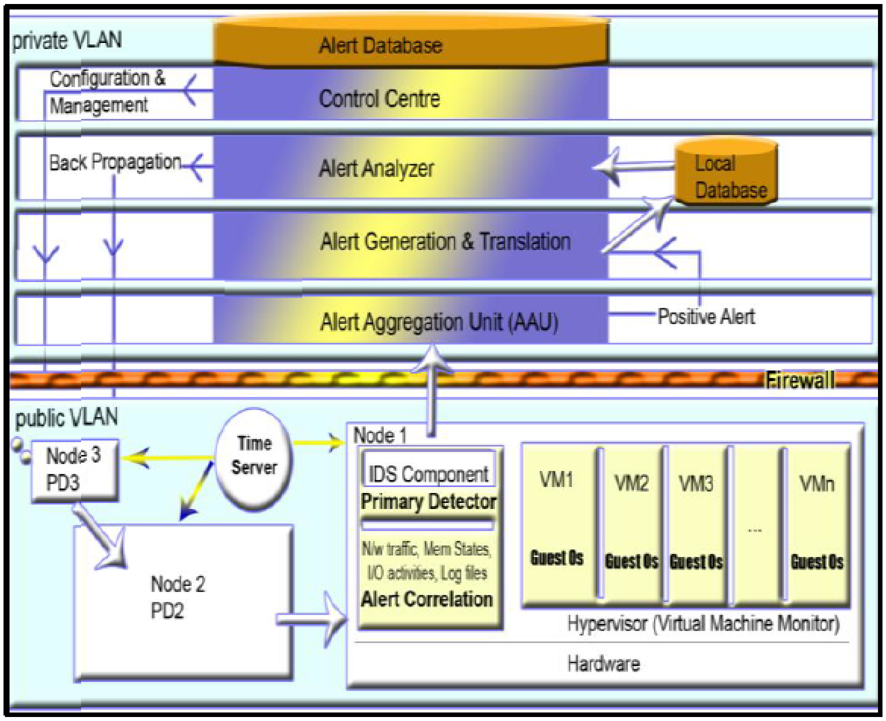

Fig. 5: Alert correlation and back propagation in multi tiers

In the proposed DCG-IDPS, the sensitive data remains rotating from one node to neighbor node in every time unit at each tier. The following algorithm describes the actions of a primary detector on recognizing an alert. The core element in the sixth tier can be the sensitive data that needs much more protection from intrusion called the alert database. Nobody can access the core except the control centre for the sake of confidence in any instance. The control centre is also restricted to access and modify the global database since valid and invalid events identified only from the data in the alert database. The hacker has to break all the other tiers of defense setup to reach the core, otherwise, the malicious activity could get blocked in the beginning itself as shown in the following flow of activities in Fig. 6.

Handling coordinated attacks: It is very difficult to detect attacks that occur in multiple domains simultaneously such as worms, stealth scans and Distributed Denial of Service (DDoS). It will be like insignificant alert but the severity can be found only after correlation. In DDoS, IDS needs to correlate alerts from multiple attack sources to a single destination but in the case of large-scale stealth scan or worm attack, there will be a correlation of single attack source responsible for numerous alerts to various destinations.

In the proposed DCG-IDPS, the sensitive data remains rotating from one node to neighbor node in every time unit at each tier. The following algorithm describes the actions of a primary detector on recognizing an alert.

Algorithm 1: Alert processing at Primary Detector (PD):

```
ap: alert priority for the current alert tp: alert threshold at PD level
if $a_p >= t_p$ then
   Call Correlation (ap) at AAU
else
   Broadcast alertTime at, Identifier for this PD (at, PD_ID) to all other
   PDs
end if
do
    if (alert matrix available in this PD)
       Call Correlation () at Primary Detector
        Exit ()
 else
    wait
  end if
while (alert matrix not available in this PD)
```

Each raw alert will be checked for its priority level. If the alert priority is alarming then immediately the alert data passed on to the next tier of alert aggregation unit for correlation else the alert generation time at and Identifier of the alert generating Primary Detector (PD_ID) broadcast to all the remaining PDs for getting previous alerts status. The previous alerts are maintained as an alert matrix M_a.

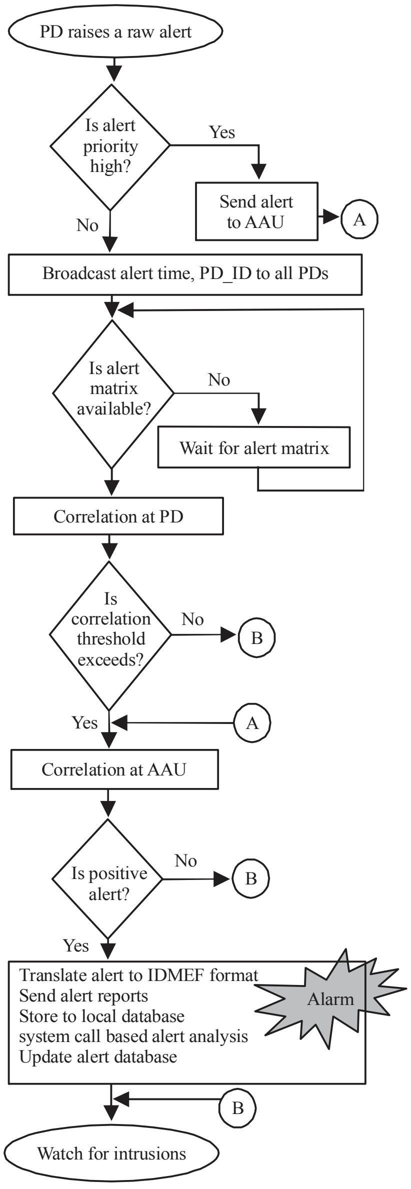

Fig. 6: Alert handling at various tiers

$$M_a = \begin{bmatrix} a11 & a12 & ...a1m \\ a21 & a23 & ...a2n \\ a31 & a32 & ...a3o... \end{bmatrix}$$

Here, rows represent the alert type (priority), columns represent the number of similar alerts raised up to the maximum of a threshold value for every alert type. In the alert matrix, m, n, o, … are the alert threshold values that are not necessarily the same. For example, a 25 specifies the 2nd priority alert recognized for the 5th time. On receiving a broadcast packet, the following algorithm details the actions of PD.

Algorithm 2; reaction of a PD on reception of a broadcast packet from PD_ID:

```
if (alert matrix Ma available) then
   if ($a_t$ not exceed time out interval) then Move alert matrix Ma to PD-ID
   end if
end if
```

Algorithm 3; Correlation at Primary Detector (PD_ID):

```
$a_p$: alert priority for the current alert
$t_p$: alert threshold at PD level
for (each alert type i = 1 to n)
   for (each alert number j = 1 to m)
         Find MP: maximum alert probability of $a_p$ after $a_{ij}$ if (MP of ap is maximum for $a_{ij}$) then
               add ap as ai,j+1
               if (j+1>tp) then
                   Call Correlation (ap) at AAU
              end if
else create a new row in Ma for new alert ap as an+1,1
end if
```

As correlation immediately performed at Primary Detector, a lot of delays avoided to find out the alert severity. If that alert is primarily severe, the alert vector with priority p (A_p) passed over to the next tier for correlation at Alert Aggregation Unit (AAU). Here, set of similar alerts get processed to a positive alert and passed over to the next tier for generation of alert reports.

Algorithm 4; Correlation at Alert Aggregation Unit (AAU):

```
$A_p$: alert vector of similar alerts {a1, a2, a3, … tp}
$t_a$: alert threshold at AAU level
s: Correlation sensitivity
Initialize Positive alert matrix P to null
for all ai in $A_p$
     for all positive alert in P
           find MP : maximum alert probability of $a_i$ after $a_j$ in $P_j$
             if MP>ta then
                for each alert ak in $P_j$
                if MP-probability between aj and Pj<s then
                  add ai with ak
                else
                  create a new Positive alert with ai
```

Now, the positive alert vector P_j gets correlated to a real alert and reported for further action through alert generator and translator components in the next tier. The IDMEF translated real alert stored in a local database for alert analysis and a final decision on intrusions.

System call analysis for malware detection: The alert analyzer component performs system call analysis for possible malware evasion from the call traces of the local database. Among a several training and detection algorithms used in a supervised learning context, linear

Table 2: Detection of distributed port scan attack-comparison with existing IDPS

Attack feature	Other IDS activity	DCG-IDPS
Even 64 scanners are not enough to detect distributed attacks in threshold based detection	Port scan will not be taken as harmful until it reaches a threshold at management component	Port scans detected immediately at the PD level correlation with a lower threshold
Outdated databases leads to high ASR	As many port scan activities go undetected, the database remains outdated	Correlation at various levels and back propagation leads to the continuously updated database
Parallel distribution leads to successful attack obfuscation	Minimal port scans will go unrecognized at sensor level or left as false positives	Correlations were done at PD level simultaneously as the port scans are going to reveal the attack immediately

Support Vector Machines (SVMs) found suitable for this defensive formation of rotating linear flow of sensitive data.

Here, it can be readily identified that the non-linear data flow promptly denotes some illegal activity. The Linear SVM algorithm separates data points into two classes with a hyperplane w^Tx. Here w defines the hyperplane learned from training data with feature vectors $x_i \in X$ and $y_i \in \{-1, 1\}$ using optimization algorithm Stochastic Gradient Descent (SGD). SGD suggests one objective function for the precisely identifying malicious process with a regularization constant α and loss function L:

$$L(t, y) = \max(0, 1\text{-}ty)$$

The objective function is:

$$E(w) = 1/p\sum_{i=1 \text{ to } p} L(y_i, w^Tx_i) + \alpha\|w\|_2$$

The process is marked as malicious if $w^Tx >$ Threshold value.

The scenario of a distributed port scans attack: A port scan attack normally sends client requests to a range of server port addresses on a host stealthily with the goal of some reconnaissance activity. That will be used by worms or malicious hackers to find an active port and weaknesses of a network. The port scan can be distributed either parallel or naïve to go undetected. Most of the commercial IDSs uses threshold based detection techniques for such port scan attacks. A port scan is said to be successful for an attacker when it goes undetected, correct port state detected and generated traffic reaches targets.

Attacker Success Rate ASR = n /T Here, n = number of ports scanned before detection, T = total number of ports to scan IDS should lower the ASR to defeat such kind of reconnaissance activities. The detection activities of commercially available IDSs compared with the proposed Padmavyuha formation in the event of a port scan attack is compared in Table 2. Figure 7 shows the step by step correlation in the case of port scan attack.

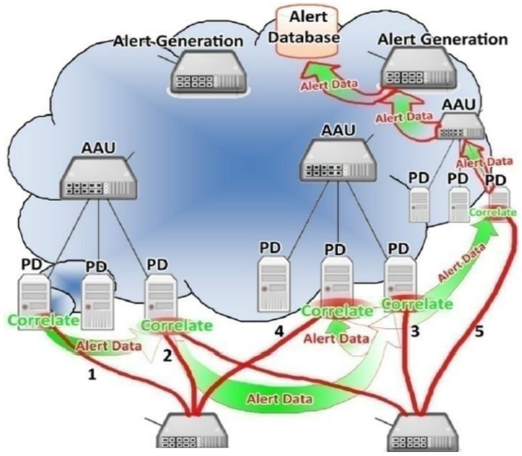

Fig. 7: Port scan attack detection

Evaluation: The evaluation of the results arrived by macro scheduling into two main modules. The first module is to achieve the Chakravyuha lab formation and to prove a reduction in alert generation delay. The second module is to apply Support Vector Machine based SGD algorithm for supervised machine learning with syscall tree analysis.

Experimental setup: The architectural framework modeled with eucalyptus 3.2.0 cloud on CentOS 6.3 as 2 clusters. Internal traffic captured by NIDS sensors with SNORT performs the role of PDs and Node controllers acting as AAUs. Cloud Controllers on independent machines generates alert reports. Local database attached to the controller for alert analysis. Central database placed separately with VLAN setup. Tcpdump and libpcap sniffer tools capture packets. The RBF kernel with gamma = 0:125 and C = 2:0 used with a window size of t = 2 sec, ST = 0:5.

RESULTS AND DISCUSSION

The experiments performed on different datasets as detailed in Table 3. Figure 8-10 show the results comparison of existing two basic approaches GCCIDS-Grid and Cloud Computing Intrusion Detection System[17], HSGAA-Heuristic Semi-Global Alignment Approach[19] with the proposed DCG IDPS framework.

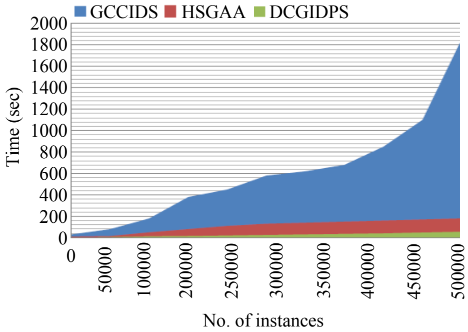

Fig. 8: Learning time

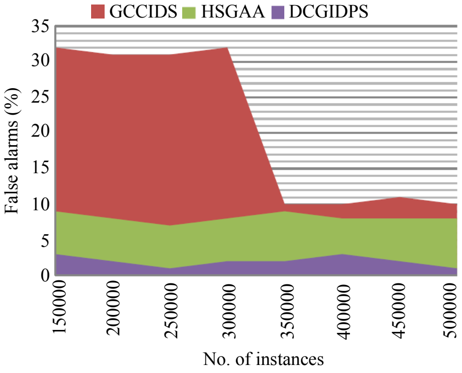

Fig. 9: False positives

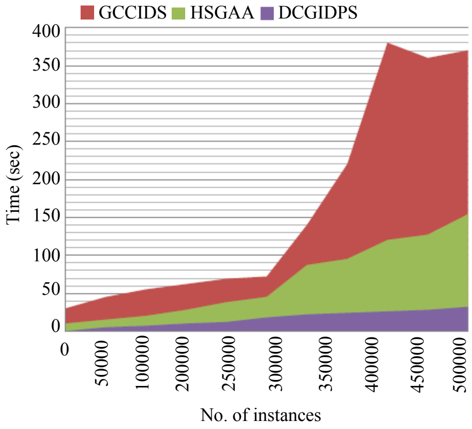

Fig. 10: Alert generation delay

Table 3: Datasets

Experiments No.	Training dataset	Test dataset
1	NSL-KDD	NSL-KDD test
2	KDD99(10%)	KDD99
3	KDD99(10%)	KDD99 test(10%)
4	ITOC	ITOC test

Table 4: Comparison of proposed architecture with arlier work

Earlier work	Dynamic	Scalable	Self-adaptive	Efficiency
Vieira *et al.*[17] GCCIDS	No	No	No	Partial
Tupakula *et al.*[19]	No	No	No	Partial
Kholidy *et al.*[19] HSGAA	Partial	Partial	Partial	No
DCG IDPS	Yes	Yes	Yes	Yes

Figure 8 reveals that the proposed system tremendously reduces alert generation delay than the others. Figure 9 compares false positive generation which is much less in DCG IDPS. In Fig. 10, DCG IDPS identified with less training time than the two approaches. Performance evaluation results with a weighted average outlined with Table 4 and Fig. 9. The results on NSL-KDD (Exp. 1) reveal that 99.52% intrusions are detected, 0.48% intrusions are true negatives, 1.27% alarms are mistaken and accuracy is 99.08%. From the results on KDD99 (Exp. 2), 99.56% intrusions are totally detected, 0.44% intrusions missed, 7.14% alarms are false and overall accuracy is 97.36%. Results on KDD99 (Exp. 3) show that almost 99.99% intrusions are detected, 0.01% intrusions are missing, 0.01% alarms are false and overall accuracy is 99.99%. Results on ITOC (Exp. 4) show that 90.53% intrusions are detected, 9.47% intrusions are missing, 14.03% alarms are false and overall accuracy is 91.5%. Weighted average results show that detection time is 32 microseconds, 99.6% intrusions are detected, 0.4% intrusions are missing, 0.22% alarms are false and overall accuracy is 99.24%. The proposed system is found to be efficient with all of these performance metrics.

From the comparison of results in existing systems, it is found that periodic alert checking causes all such delays in taking response actions. In Table 5, our proposed architecture compared with the earlier work in the terms of dynamic, scalable, self-adaptive and Efficiency. Proposed algorithm seems to be highly efficient for further incident response management. After alert generation process has completed without delay, the more time is available for quantitative and qualitative risk analysis. Quantitative risk uses Annual Loss expectancy (ALE) to determine the amount of loss that is associated with a particular risk. Risk = Probability of loss X value of loss.

Then we can also take other countermeasures based on the expected risk as the following. (Attack Success+ Criticality)-(Countermeasures) = Risk This process facilitates supervised learning of SVM, since, the non-linear flow of system calls defines the malicious event taking place.

The framework of DCGIDPS architecture can be expanded as per the nature of the network in which the system is deployed. The working of the system will vary corresponding to the nature of the network.

Table 5: Performance evaluation

Experiment No.	Total detections (%)	True negatives (%)	True positives (%)	False positives(%)	Accuracy (%)
Exp 1 (NSL KDD)	99.52	0.48	98.73	1.27	99.080
Exp 2 (KDD99 10%)	99.56	0.44	92.86	7.14	97.360
Exp 3 (KDD99 10%-test)	99.99	0.01	99.99	0.01	99.988
Exp 4 (ITOC)	90.53	9.47	85.97	14.03	91.500
Wt. average	99.60	0.40	99.78	0.22	99.240

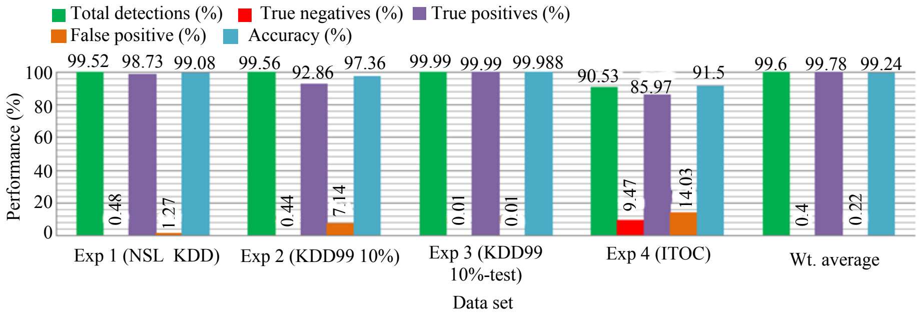

Fig. 11: Performance evaluation

CONCLUSION

An efficient multi-tier defensive formation proposed in this study for the VLAN of management IDPS Components with ultimate security requirements. The formation narrated in the Indian epic Mahabharata as an impregnable strategy was already analyzed by many countries for their military formation. The multi tiers of the incident processing make the model to generate alerts with likely less number of false positives. As every alert immediately correlated with its occurrence, the alert generation delay tremendously reduced. The revolving sensitive data in every component on each tier make the model a unique one. However, this data movement introduces extra overhead on regular IDS activity. The model can be further explored for reducing such complexity as a future enhancement.

REFERENCES

01. Furht, B., 2010. Cloud Computing Fundamentals. In: Handbook of Cloud Computing, Furht, B. and A. Escalante (Eds.). Springer, Boston, Massachusetts, pp: 3-19.
02. Scarfone, K. and P. Mell, 2007. Guide to Intrusion Detection and Prevention Systems (IDPS). NIST. Spec. Publ., 800: 94-127.
03. Zhou, C.V., C. Leckie and S. Karunasekera, 2010. A survey of coordinated attacks and collaborative intrusion detection. Comp. Security, 29: 124-140.
04. Man, N.D. and E.N. Huh, 2012. A Collaborative Intrusion Detection System Framework for Cloud Computing. In: Proceedings of the International Conference on IT Convergence and Security 2011, Kim, K.J. and S.J. Ahn (Eds.)., Springer, Dordrecht, Netherlands, pp: 91-109.
05. Roschke, S., F. Cheng and C. Meinel, 2009. An extensible and virtualization-compatible IDS management architecture. Proceedings of the 2009 5th International Conference on Information Assurance and Security Vol. 2, August 18-20, 2009, IEEE, Xi'an, China, pp: 130-134.
06. Bakshi, A. and Y.B. Dujodwala, 2010. Securing cloud from DDOS attacks using intrusion detection system in virtual machine. Proceedings of the 2nd International Conference on Communication Software and Networks (ICCSN'10), February 26-28, 2010, IEEE, Singapore, ISBN:978-1-4244-5726-7, pp: 260-264.
07. Lo, C.C., C.C. Huang and J. Ku, 2010. A cooperative intrusion detection system framework for cloud computing networks. Proceedings of the 2010 39th International Conference on Parallel Processing Workshops, September 13-16, 2010, IEEE, San Diego, California, pp: 280-284.
08. Mazzariello, C., R. Bifulco and R. Canonico, 2010. Integrating a network ids into an open source cloud computing environment. Proceedings of the 2010 6th International Conference on Information Assurance and Security, August 23-25, 2010, IEEE, Atlanta, Georgia, pp: 265-270.
09. Brown, D.J., B. Suckow and T. Wang, 2002. A survey of intrusion detection systems. Ph.D Thesis, Department of Computer Science, University of California, San Diego, California.
10. Dutkevych, T., A. Piskozub and N. Tymoshyk, 2007. Real-time intrusion prevention and anomaly analyze system for corporate networks. Proceedings of the 4th IEEE Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, September 6-8, 2007, IEEE, New York, USA., ISBN:978-1-4244-1347-8, pp: 599-602.

11. Zhengbing, H., S. Jun and V.P. Shirochin, 2007. An intelligent lightweight intrusion detection system with forensics technique. Proceedings of the 2007 4th IEEE Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, September 6-8, 2007, IEEE, Dortmund, Germany, pp: 647-651.
12. Garfinkel, T. and M. Rosenblum, 2003. A virtual machine introspection based architecture for intrusion detection. Proceedings of the Network and Distributed Systems Security Symposium, February 6-7, 2003, San Diego, California, USA., pp: 191-206.
13. Vieira, K., A. Schulter, C.B. Westphall and C.M. Westphall, 2010. Intrusion Detection for grid and cloud computing. IT Prof., 2: 38-43.
14. Dastjerdi, A.V., K.A. Bakar and S.G.H. Tabatabaei, 2009. Distributed intrusion detection in clouds using mobile agents. Proceedings of the 3rd International Conference on Advanced Engineering Computing and Applications in Sciences, October 11-16, 2009, IEEE, Johor Baru, Malaysia, ISBN:978-1-4244-5082-4, pp: 175-180.
15. Guan, Y. and J. Bao, 2009. A CP intrusion detection strategy on cloud computing. Proceedings of the International Symposium on Web Information Systems and Applications, May 22-24, 2009, South China University of Technology, Guangzhou, China, ISBN:978-952-5726-01-5, pp: 84-87.
16. Modi, C., D. Patel, B. Borisaniya, H. Patel and A. Patel *et al*., 2013. A survey of intrusion detection techniques in cloud. J. Netw. Comput. Appl., 36: 42-57.
17. Vieira, K., A. Schulter, C. Westphall and C. Westphall, 2009. Intrusion detection techniques in grid and cloud computing environment. IT Professional, 99: 1-1.
18. Tupakula, U., V. Varadharajan and N. Akku, 2011. Intrusion detection techniques for infrastructure as a service cloud. Proceedings of the 2011 IEEE 9th International Conference on Dependable, Autonomic and Secure Computing, December 12-14, 2011, IEEE, Sydney, Australia, pp: 744-751.
19. Kholidy, H.A. and F. Baiardi, 2012. CIDS: A framework for intrusion detection in cloud systems. Proceedings of the 2012 9th International Conference on Information Technology-New Generations, April 16-18, 2012, IEEE, Las Vegas, Nevada, pp: 379-385.
20. Cepheli, O., S. Buyukcorak and K.G. Kurt, 2016. Hybrid intrusion detection system for DDoS attacks. J. Electr. Comput. Eng., Vol. 2016, 10.1155/2016/1075648.
21. Riquet, D., G. Grimaud and M. Hauspie, 2012. Large-scale coordinated attacks: Impact on the cloud security. Proceedings of the 2012 6th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, July 4-6, 2012, IEEE, Palermo, Italy, pp: 558-563.
22. Canzanese, R., S. Mancoridis and M. Kam, 2015. System call-based detection of malicious processes. Proceedings of the 2015 IEEE International Conference on Software Quality, Reliability and Security, August 3-5, 2015, IEEE, Vancouver, Canada, pp: 119-124.
23. Zhang, T., 2004. Solving large scale linear prediction problems using stochastic gradient descent algorithms. Proceedings of the 21st International Conference on Machine Learning, July 2004, ACM, Banff, Alberta, pp: 116-124.