Defence Science Journal, Vol. 67, No. 6, November 2017, pp. 645-653, DOI : 10.14429/dsj.67.11118

Impregnable Defence Architecture using Dynamic Correlation-based Graded Intrusion Detection System for Cloud

K. Umamaheswari[*] and S. Sujatha[#]

[*]*Research and Development Centre, Bharathiar University, Coimbatore - 641 046, India*
[#]*Department of Computer Science, Bharathi Women's College, Chennai - 600 108, India*
[*]*E-Mail: uma.tvr1981@gmail.com*

ABSTRACT

Data security and privacy are perennial concerns related to cloud migration, whether it is about applications, business or customers. In this paper, novel security architecture for the cloud environment designed with intrusion detection and prevention system (IDPS) components as a graded multi-tier defense framework. It is a defensive formation of collaborative IDPS components with dynamically revolving alert data placed in multiple tiers of virtual local area networks (VLANs). The model has two significant 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 defence formation facilitates malware detection with linear support vector machine- stochastic gradient descent (SVM-SGD) statistical algorithm. It requires little computational effort to counter the distributed, co-ordinated attacks efficiently. The framework design, then, takes distributed port scan attack as an example for assessing the efficiency in terms of reduction in alert generation delay, the number of false positives and learning time through comparison with existing techniques is discussed.

Keywords: Chakravyuha or Padmavyuha; Intrusion detection and prevention systems; IDPS; Multi-tier defence framework; SVM – SGD; System call analysis; Virtual local area networks; VLAN

1. INTRODUCTION

It is becoming a scarce sight in the information technology world for enterprise owned data centers, as organisations move their business to outsource their infrastructure requirements with the cloud provider communities. There is a fundamental shift in the security boundary for the enterprise's sensitive data. Hence, there is an increased need for a ubiquitous security approach.

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, smartphones, 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 cloud security. Intrusion detection system (IDS) is one of such tools for alerting any sign of intrusion activities at the virtual machine level of virtualised cloud[2]. Intrusion detection and prevention systems (IDPS) include all protective actions that identification of possible incidents, analysing log information of such incidents, how to block them in the beginning itself and generate reports for the concern of security personnel[3]. It is much more important to secure IDPS components since it is the primary target of attackers who try to prevent the IDPSs functioning of detecting 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 networks[5].

VLANs pave way for logically segregating network traffic on all management communications of the IDPS components. 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 authorised 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.

The proposed model of defence framework arranges intrusion detection components in a maze-like structure so as to capture and dynamically correlate unknown attacks as early as possible.

Received : 28 January 2017, Revised : 21 August 2017
Accepted : 11 September 2017, Online published : 06 November 2017

2. STATE-OF-ART IN CLOUD SECURITY

While applying IDPS for cloud security, a variety of traditional techniques are available such as signature-based detection, anomaly detection, artificial intelligence (AI) based detection, etc. Signature-based intrusion detection can detect known attacks only. Roschke[6], *et al.* suggested an extensible architecture for integrating VM management and IDS management. Bakshi[7], *et al.* proposed an approach to secure cloud from DDOS Attacks using intrusion detection system in a virtual machine. Lo[8], *et al.* used signature based detection for building a co-operative IDPS in the cloud. C. Mazzariello[9], *et al*. integrated a network IDS into an open source cloud computing environment. Anomaly or behavioural detection alerts anomalous events by comparing with normal behaviour[10]. This approach is efficient in the sense that it lowers false alarms for both known and new attacks. This technique can be used for the cloud to detect unknown attacks at different levels[11]. Zhengbing[12], *et. al.* proposed a lightweight IDS with forensic techniques of anomaly-based detection. Garfinkel[13] suggested a virtual machine introspection based architecture for intrusion detection. Various anomaly-based intrusion detection techniques proposed for both grid and cloud computing environments[14,15]. Dastjerdi[16], *et al.* proposed a technique for distributed intrusion detection in the cloud using mobile agents. But a large number of events in the cloud make it tougher 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[17].

Hybrid techniques combine the advantages of more than one technique. Man[4] proposed a technique of arranging the IDPS components in a hierarchical manner for handling large-scale coordinated attacks. The setup was a collaboration of IDPS components located in various cloud provider networks. In ultra-secure-network- architecture[5], the IDPS components arranged in various tiers separated into distinct demilitarised zones. This model is vulnerable to some incidents aiming at the sensitive data on IDPS components.

Kleber[18], *et al.* presented a hybrid intrusion detection system for the cloud that can detect only selective kind of attacks. Hence the system cannot be deployed in a real-time distributed environment. Tupakula[19], *et al.* 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 synchronisation characteristics are not applicable to the system. Kholidy[20], *et al*. framework does not detect the intrusions in a faster manner; The system can handle large scale, dynamic data only partially. Some more systems[21] handle a few of the renowned attacks efficiently. Riquet[22], *et al.* 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[23]. Linear support vector machine (SVM) based stochastic gradient descent (SGD) algorithm suitably assists supervised learning of unknown malware detection[24]. Having a third party administrator for securing data can be an effective alternative but the cloud user cannot ensure data confidentiality[25]. Various collaborative IDS techniques surveyed by Vasilomanolakis[26], *et al.* have not considered unknown attacks.

The proposed system is a hierarchical defense framework with protective measures against the vulnerabilities present in existing systems. The components isolated from common data traffic through positioning in VLAN tiers. The most confident, alert data securely move in such a way that its position cannot be predicted in advance by intruders. System call analysis on the flow of alert data helps to learn unknown malware effectively than any other existing technique.

3. PROPOSED SYSTEM

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

3.1 Inspirations

The defensive framework for positioning management IDPS components in a 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[27] as shown in Fig. 1.

Logically, a *Chakravyuha* should be a multi-layered circular labyrinthine maze where each of the layers is rotating in same or opposite direction, in which weak and strong warriors are strategically placed, and each of the layers is presented with possible openings which are closely guarded by one of the main highly ranked warriors and his personal troops[28]. The rotating nature nullified the plans from the opponents, which

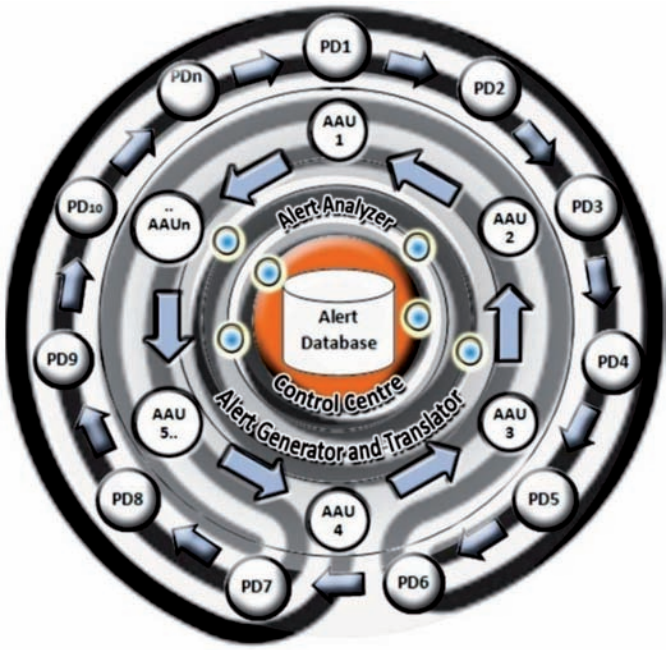

Figure 1. Multi-tier defence formation of management IDPS components in cloud.

they might have devised against any particular warrior within *Chakravyuha* and thus confused them off their strategies. 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 entry of intruders.

3.2 Key Considerations

Often it can be found that key innovative techniques in research start their journey from the technology of defence. Table 1 shows how the existing issues in applying Intrusion detection system for cloud can be solved with *Chakravyuha* formation. In the proposed system, increased security is obtained by moving the sensitive attack data from one component to another within every tier so that the sensitive data on attacks readily available for correlation. Here, one time server is used to synchronise the data movement among IDPS components.

4. DYNAMIC CORRELATION BASED GRADED INTRUSION DETECTION AND PREVENTION SYSTEM

The outer tier of dynamic correlation based graded intrusion detection and prevention system (DCG-IDPS) is formed by all the sensors or the agents defined as host level (HIDS) or network level (NIDS) that primarily recognise any malicious event called primary detectors (PDs). The PDs collect and analyse 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 means of starting correlation of raw alerts in the primary detector (PD) level itself with an 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). In AAU, the next level of the threshold is 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 intrusion detection message exchange format (IDMEF). The translation performed by extracting the necessary data and stored in a local database. In the fourth tier, alert analyser 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 of 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 control centre in the fifth tier is the management component of information exchange. It acts based on user 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. 2.

The core element in the sixth tier can be the sensitive data that need 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 the defence setup to reach the core, otherwise, the malicious activity could get blocked at the beginning itself as shown in the above flow of activities in Fig. 3.

4.1 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 the single attack source responsible for numerous alerts to various destinations.

Algorithm 1: Alert processing at Primary Detector (PD)
a_p: alert priority for the current alert
t_p: alert threshold at PD level
if $a_p >= t_p$ then
Call Correlation (a_p) at AAU;

Table 1. How Chakravyuha fits in IDPS architecture

Threat Issues	Existing IDPS System	*Chakravyuha* framework
Alert data remain idle in a node until a time limit reaches.	Almost the hackers' activity has 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 the alert database got hacked, 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.

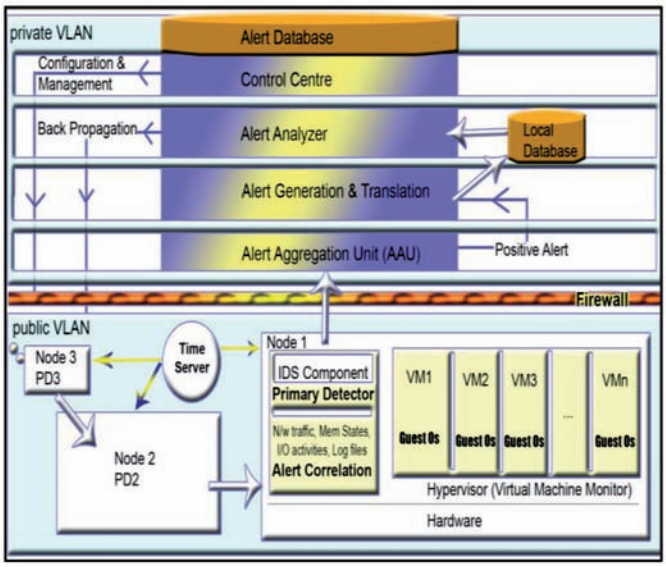

Figure 2. Alert correlation and back propagation.

else
Broadcast alertTime a_t, Identifier for this PD (a_t, 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);

In the proposed DCG-IDPS, the sensitive data remain rotating from one node to the neighbor node in every time unit at each tier. Algorithm 1 describes the actions of a primary detector on recognising an alert. 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 the alert aggregation unit for correlation else the alert generation time a_t and identifier of the alert generating primary detector (PD_ID) broadcast to all the remaining PDs for getting previous alert status. The previous alerts are maintained as an alert matrix M_a.

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

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, a25 specifies the 2nd priority alert recognised for the 5th time. Algorithm 2 and 3 details the actions of a PD on receiving a broadcast packet.

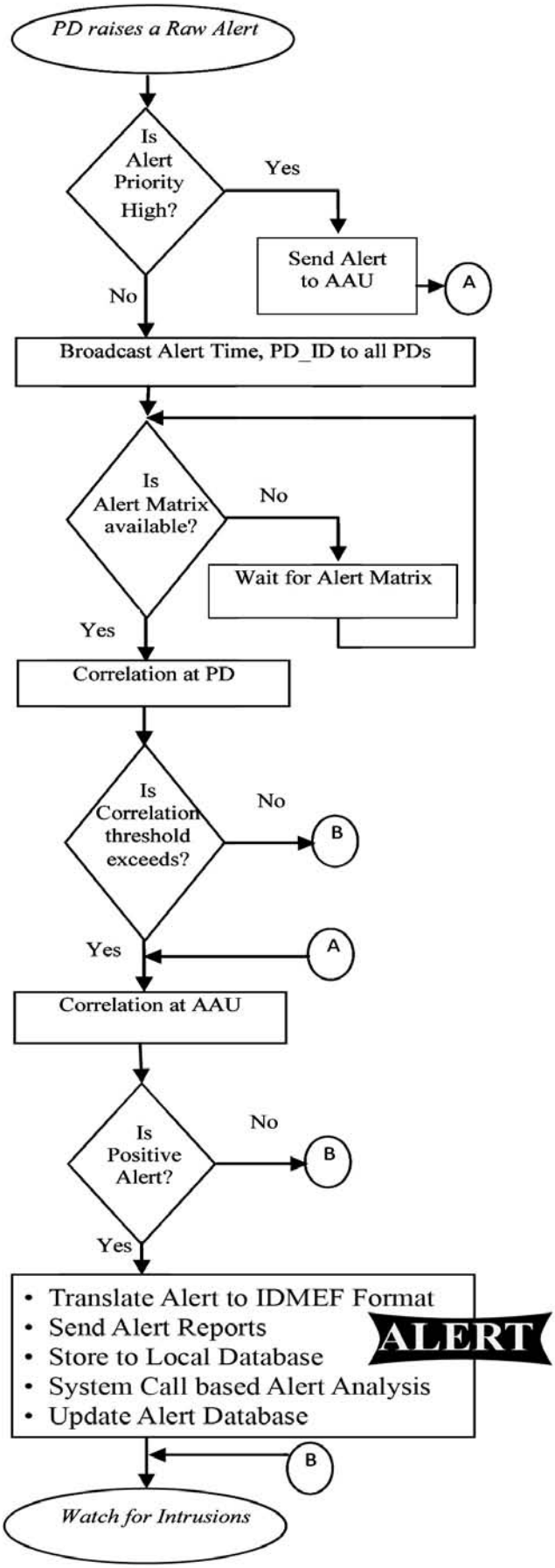

Figure 3. Alert handling at various tiers.

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

if (alert matrix M_a available) then
if (a_t not exceed time out interval) then
Move alert matrix M_a 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 a_p is maximum for a_{ij}) then
add a_p as $a_{i,j+1}$;
if (j+1>tp) then
Call Correlation (a_p) at AAU;
end if;
else
create a new row in M_a for new alert a_p as $a_{n+1,1}$;
end if;

If that alert is primarily severe, the alert vector with priority p (Ap) passed over to the next tier for correlation at the alert aggregation unit (AAU) as detailed in Algorithm 4. Here, a set of similar alerts get processed to a positive alert and passed over to the next tier for the generation of alert reports. Now the positive alert vector Pj gets correlated to a real alert and reported for further action through alert generator & 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.

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

A_p: alert vector of similar alerts {a_1, a_2, a_3, … t_p}
t_a: alert threshold at AAU level
s: Correlation sensitivity
Initialize Positive alert matrix P to null
for all a_i 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>t_a then
for each alert a_k in Pj
if MP-probability between aj and Pj<s then
add ai with ak;
else
create a new Positive alert with ai;

4.2 System Calls Analysis for Malware Detection

The alert analyser component performs system call analysis for possible malware evasion from the call traces of the local database. Among a lot of training and detection algorithms used in a supervised learning context, linear 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 hyper plane w_x^T. Here *w* defines the hyperplane learned from training data with feature vectors $x_i \in X$ and $y_i \in \{-1,1\}$ using optimisation algorithm stochastic gradient descent (SGD). SGD suggests one objective function (2) for precisely identifying the malicious process with a regularisation constant α and loss function L shown in Eqn (1) with *p* number of training samples. DCGIDPS assigns a threshold value γ based on the number of training samples taken in the hyper plane.

$$L(t, y) = \max(0, 1 - ty) \quad (1)$$

The objective function is

$$E(w) = \frac{1}{p}\sum_{i=1}^{p} L(yi, wTxi) + a\|w\|_2 \quad (2)$$

The process is marked as *malicious* if $w_x^T > \gamma$.

4.3 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. Most of the commercial IDSs use 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.

$$\text{Attacker Success Rate } ASR = \frac{n}{T} \quad (3)$$

Here *n* = number of ports scanned before detection
and *T* = total number of ports to scan

The 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 4 shows the step by step correlation in the case of port scan attack.

5. 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.

5.1 Experimental Setup and Materials

The architectural framework modelled 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.

The experiments performed on different datasets, as detailed in Table 3. Cloud controllers on independent machines generate 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. Optunity tool has used to optimize hyper parameters for the support vector machine classifier (SVC) in scikit-

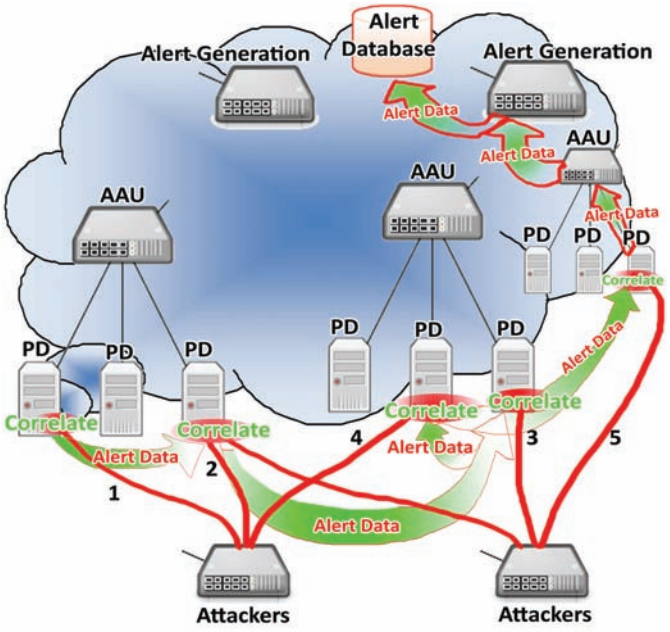

Figure 4. Port scan attack detection.

learn. Optunity's conditional hyper parameter optimisation feature used here to optimise over all RBF kernel functions and their associated hyper parameters at once for the principal datasets.

5.2 Results

Figures 5(a) - 5(c) show the results comparison of existing two basic approaches GCCIDS - Grid and Cloud Computing Intrusion Detection System[18], HSGAA - Heuristic Semi-Global Alignment Approach[19] with the proposed DCG IDPS framework. In Fig. 5(a), DCG IDPS identified with less training time than the two approaches. Figure 5(b) compares false positive generation, which is much less in DCG IDPS. Figure 5(c) reveals that the proposed system tremendously reduces alert generation delay than the others. Performance evaluation results with a weighted average outlined in Table 4. The results on NSL-KDD (Exp1) reveal that 99.52 per cent intrusions are detected, 0.48 per cent intrusions are true negatives, 1.27 per cent alarms are mistaken and accuracy is 99.08 per cent.

From the results on ISCX 2012 (Exp 2), 99.56 per cent intrusions are totally detected, 0.44 per cent intrusions missed, 7.14 per cent alarms are false and overall accuracy is 97.36 per cent. Results on KDD99 (Exp 3) show that almost 99.99 per cent intrusions are detected, 0.01 per cent intrusions are missing, 0.01 per cent alarms are false and overall accuracy

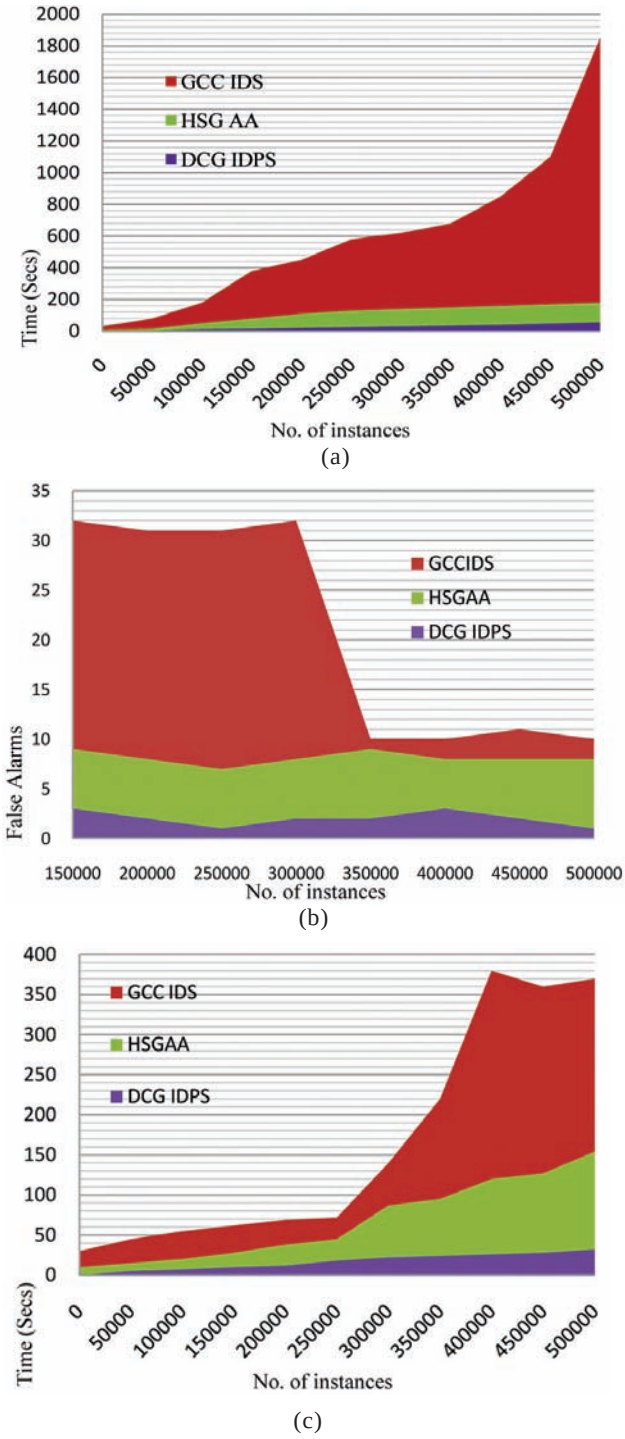

Figure 5. Comparative analysis of results: (a) Learning time (b) False positives generation, and (c) Alert generation delay.

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 (Eqn 3)	As many port scan activities go undetected, the database remains outdated.	Correlation in various levels and back propagation leads to the continuously updated database.
Parallel distribution leads to successful attack obfuscation	Minimal port scans will go unrecognised 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.

Table 3. Datasets

Exp. No.	Training dataset	Test dataset
Exp 1	NSL-KDD	NSL-KDD test
Exp 2	ISCX 2012	ISCX 12 test
Exp 3	KDD99 (10 per cent)	KDD99 test (10 per cent)
Exp 4	ITOC	ITOC test

the expected risk as the following.

(Attack Success + Criticality) − (Countermeasures) = Risk

The proposed system comparatively eliminates a lot of alert generation delay at the same time of producing true alarms with reduced learning time.

Hence, it is proposed as a suitable IDS architecture for the cloud environment than any other existing methodology.

Table 4. Performance Evaluation

Experiment No.	Total detections (per cent)	True negatives (per cent)	True positives (per cent)	False positives (per cent)	Accuracy (per cent)
Exp 1 (NSL KDD)	99.52	0.48	98.73	1.27	99.08
Exp 2 (ISCX 2012)	99.56	0.44	92.86	7.14	97.36
Exp 3 (KDD99 10 per cent - Test)	99.99	0.01	99.99	0.01	99.988
Exp 4 (ITOC)	90.53	9.47	85.97	14.03	91.5
Wt. Average	99.6	0.4	99.78	0.22	99.24

is 99.99 per cent. Results on ITOC (exp4) show that 90.53 per cent intrusions are detected, 9.47 per cent intrusions are missing, 14.03 per cent alarms are false and overall accuracy is 91.5 per cent. Weighted average results show that detection time is 32 microseconds, 99.6 per cent intrusions are detected, 0.4 per cent intrusions are missing, 0.22 per cent alarms are false and overall accuracy is 99.24 per cent. The proposed system is found to be efficient with all of these performance metrics.

6. DISCUSSIONS

In earlier methods, correlation performed only in higher level layers to ensure the clarity of an incident. This incurs a delay in the alert generation at higher level components. From the comparison of results with 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. The 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

Table 5. Comparison with earlier work

Earlier work	Dynamic	Scalable	Self-adaptive	Efficiency
Kleber[18], *et al.* GCCIDS	No	No	No	Partial
Tupakula[19], *et al*.	No	No	No	Partial
Kholidy[20], *et al.* HSGAA	Partial	Partial	Partial	No
DCG IDPS	Yes	Yes	Yes	Yes

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.

7. CONCLUSION

An efficient multi-tier defensive formation proposed in this paper 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 analysed by many countries for their military formation. The multi-tier of the incident processing make the model to generate alerts with likely less number of false positives. As every alert correlated immediately 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. Furthermore, as the proposed system improves the functionality of IDS components in finding unknown threats and vulnerabilities, the cloud environment will be observed more secure than ever before. 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

1. Furht, B. Chapter I - Cloud Computing fundamentals. *In* Handbook of cloud computing. Springer Science & Business Media, LLC, USA, 2010, pp. 3-19. doi:10.1007/978-1-4419-6524-0.
2. Scarfone, Karen A. & Mell, Peter M. Guide to intrusion detection and prevention systems (IDPS). Computer Security Resource Center, National Institute of Standards and Technology Special Publication (NIST SP), 2007, pp. 800-94.
3. Zhou, C.V.; Leckie, C. & Karunasekera, S. A survey of coordinated attacks and collaborative intrusion detection.

Computers Security, 2010, **29**(1), 124-140.
doi:10.1016/j.cose.2009.06.008

4. Nguyen, Doan Man & Eui-Nam, Huh. A collaborative intrusion detection system framework for cloud computing. *In* Proceedings of the International conference on IT Convergence and Security, Lecture Notes in Electrical Engineering, Springer, Dordrecht, 2011, **120**(8), pp. 91-107.
doi: 10.1007/978-94-007-2911-7_8
5. The ultra-secure network architecture. http://rsmus.com/what-we-do/services/risk-advisory/the-ultra-secure-network-architecture.html (Accessed on 25 August 2017).
6. Roschke, S.; Feng, C. & Meinel, C. An extensible and virtualization compatible ids management architecture. *In* Fifth International Conference on Information Assurance and Security, 2009, **2**, 130-134.
doi: 10.1109/IAS.2009.151
7. Bakshi, A. & Yogesh, B. Securing cloud from DDOS attacks using intrusion detection system in virtual machine. *In* 2nd International Conference on Communication Software and Networks, 2010, 260-264.
doi: 10.1109/ICCSN.2010.56
8. Lo, C.C.; Huang, C.C. & Ku, J. Cooperative intrusion detection system framework for cloud computing networks. *In* 39th IEEE International Conference on Parallel Processing Workshops (ICPPW), San Diego, 2010, 280-284.
9. Mazzariello, C.; Bifulco, R. & Canonoco, R. Integrating a network IDS into an open source cloud computing. *In* Sixth International conference on Information Assurance and Security (IAS), Atlanta, USA, 2010, 265- 270.
doi: 10.1109/ISIAS.2010.5604069
10. Brown, D.J.; Suckow, B. & Wang, T. A survey of intrusion detection systems. department of computer science. University of California, San Diego, 2002.
11. Dutkevyach, T.; Piskozub, A. & Tymoshyk, N. Real-time intrusion prevention and anomaly analyze system for corporate networks. *In* 4th IEEE Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, 2007, 599-602.
doi: 10.1109/IDAACS.2007.4488491
12. Zhengbing, H.; Jun, S. & Shirochin, V.P. An intelligent lightweight intrusion detection system with forensic technique. *In* 4th IEEE Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, Dortmund, Germany, 2007, 647-651.
doi**:** 10.1109/IDAACS.2007.4488501
13. Garfinkel, T. & Rosenblum, M. A virtual machine introspection based architecture for intrusion detection. *In* Proceeding of Network and Distributed Systems Security Symposium, 2003, pp. 191-206.
doi: 10.1.1.11.8367
14. Kene, Snehal G. & Theng, Deepti P. A review on intrusion detection techniques for cloud computing and security challenges. *In* IEEE sponsored 2nd International Conference on Electronics and Communication Systems (ICECS '2015), Coimbatore, India. 2015, 227-232.
doi: 10.1109/ECS.2015.7124898.
15. Guan, Y. & Bao, J.A.C.P. Intrusion detection strategy on cloud computing. *In* International Symposium on Web Information Systems and Applications (WISA), 2009, 84–87.
doi: 10.1.1.402.9747
16. Dastjerdi, K.A.B.A.V. & Tabatabaei, S.G.H. Distributed intrusion detection in clouds using mobile agents. *In* 3rd International Conference on Advanced Engineering Computing and Applications in Sciences, 2009, 175 – 180.
doi: 10.1109/ADVCOMP.2009.34
17. Modi, C.; Patel, D.; Patel, H.; Borisaniya, B.; Patel, A. & Rajarajan, M. A survey of intrusion detection techniques in cloud. *J. Network Comput. Appli.*, **36**(1), 2013, 42-57.
doi: 10.1016/j.jnca.2012.05.003
18. Vieira, K.; Schulter, A.; Westphall, C.B. & Westphall, C.M. Intrusion detection for grid and cloud computing. *IEEE J.: IT Professional*, 2010, **4**, 38–43.
doi: 10.1109/MITP.2009.89 · Source: IEEE Xplore
19. Tupakula, U.; Varadharaja, V. & Akku, N. Intrusion detection techniques for infrastructure as a service cloud. *In* the Proceedings of 9th IEEE International Conference on Dependable, Autonomic and Secure Computing, Sydney, 2011, 744–751.
doi: 10.1109/DASC.2011.128
20. Kholidy, Hisham A. & Baiardi, F. CIDS: A framework for intrusion detection in cloud systems. *In* the Proceedings of 9th IEEE International Conference on Information Technology-New Generations, Las Vegas, 2012, 379–85.
doi: 10.1109/ITNG.2012.94
21. Ozge, Cepheli; Saliha, Buyukcorak & GuneG, Karabulut Kurt. Hybrid intrusion detection system for DDoS attacks. *J. Electr. Comput. Eng.*, 2016, **2**, 1-8.
doi: 10.1155/2016/1075648
22. Riquet, Damien; Grimaud, Gilles & Hauspie, M. Large-scale coordinated attacks: Impact on the cloud security *In* Proceedings of the 6th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing IMIS '12, 2012, 558-563.
doi: 10.1109/IMIS.2012.76
23. Canzanese, Raymond & Mancoridis, Spiros. System call-based detection of malicious processes. *In* International conference on quality, reliability, and security. 2015, 119-124.
doi: 10.1109/QRS.2015.26
24. Zhang, Tong. Solving large scale linear prediction problems using stochastic gradient descent algorithms. *In* Proceedings of the 21st International Conference on Machine Learning, Banff, Canada. 2004,
doi: 10.1145/1015330.1015332
25. Ramachandran, B. & Subramaniam, K. Multilevel security framework based resource sharing using bilinear mapping in cloud environment. *Int. J. Intelli. Eng. Sys.*, 2017, **10(**3), 347-354.
doi: 10.22266/ijies2017.0630.39
26. Vasilomanolakis, Emmanouil; Karuppayah, Shankar;

Mühlhäuser, Max & Fischer, Mathias. Taxonomy and survey of collaborative intrusion detection. *ACM Comput. Surv. (CSUR).*, 2015 **47**(4). doi: 10.1145/2716260

27. Padmavyuha. https://en.wikipedia.org/wiki/Padma vyuha (Accessed on 22 June 2015).
28. Vyas, Ved & Dutt, M.N. Mahabharata : Sanskrit text with english translation. Parimal Publications, 2013, **5**. ISBN: 8171101966

CONTRIBUTORS

Mrs K. Umamaheswari obtained her Master's from Bharathidasan University, Tiruchirappalli, in 2004. Currently pursuing her PhD at Research and Development Centre, Bharathiar Univerisity, Coimbatore, India. Her areas of research include :Cloud security, virtualisation, machine learning, and data mining.

Her contribution in the current study includes design and development of the architectural model, system call analysis evaluation and its implementation in the real time cloud environment.

Dr S. Sujatha received her MSc (Computer Science) from Anna University, Chennai, in 2002. Obtained her PhD from Department of Mathematics, Anna University, Chennai, in 2009. Currently working as an Assistant Professor in Bharathi Women's College(A), Chennai, Tamil Nadu, India. Her current area of interest includes : Information and network security, cryptography, MANETs, soft computing and cloud computing.

Her contribution in the current study includes analysis of the previous methods and performance investigation on each stage of the proposed method.