NJESR/ Feburary2023/Volume5/Issue 2 E-ISSN-2582-5836
DOI-10.53571/NJESR.2023.5.2.33-46

Intrusion Detection And Prevention For Cloud Security

Jiby P.Joseph[*1], Susan Dilna Peter[*2]
Lecturer, Teacher
Department Of Computer Engineering[*1]
Government Polytechnic College
Perumbavoor
Department Of Computer Science[*2]
Don Bosco Senior Secondary School
Vaduthala, Emakulam

(Received:6January2023/Revised:30January2023/Accepted:10Feburary2023/Published:21Feburary2023)

Abstract

Organizations are turning to cloud computing as a result of the rapid expansion of resources and rising infrastructure costs. High performance, effective resource utilization, and on-demand resource availability are all features of cloud computing. However, the cloud environment is susceptible to a variety of intrusion attacks that involve the creation of backdoors and the installation of malicious software. When businesses host important and critical data in a cloud environment, the security of the underlying technologies becomes critical. Intrusion Detection Systems, or IDS, are a protective layer for cloud environments. This survey paper aims to examine proposed cloud-based IDS methods. The first step in achieving this goal is to identify each technique's limitations and distinctive features. The criteria for evaluating IDS architectures are established in the second step. The basic characteristics of the cloud serve as the basis for the criteria used in this paper. A comparison of various existing intrusion detection methods to the criteria is the next step. The final step is to talk about the problems and open questions from the evaluation that make it hard to implement IDS in a cloud environment.

Keywords: Intrusion Detection Systems, Cyber-Security, Cloud Computing, Comparative Analysis, Open Issues

Introduction

Due to its low price and pay-as-you-go model, cloud computing is a new technology that is being adopted by businesses of all sizes. With its one-of-a-kind and widespread capabilities, it has revolutionized the IT industry. The company prefers the cloud because it eliminates the need for costly infrastructure and ongoing upkeep. Software as a service (such as Google Apps[1]), platform as a service (such as Google App Engine[2] and Microsoft Azure[3]), and infrastructure as a service (such as Amazon Web Service, Eucalyptus, and Open Nebula) are

its three service models. The cloud can offer scalability, elasticity, ease of use, and on-demand network access to a shared pool of configurable computing resources thanks to virtualization. The service-oriented architecture of the cloud computing paradigm has resulted in a significant shift in the delivery and administration of services.

Any computing environment employs intrusion detection methods as a security measure. The fundamental objective is to identify any malicious activity well before it can cause significant harm. The general idea is to find attacks by looking at system artifacts (like log files and process lists) or or by monitoring network activity. Signature-based detection and anomaly-based detection are the two primary methods utilized. By defining patterns in known attack signatures, signature-based detection works. The system is flagged as an intrusion if it is found to be processing any code that is similar to those signatures. Anomaly-based detection, on the other hand, works by looking at how the system is used. A system's profile is initially created by recording typical activities (such as setting bandwidth usage thresholds). An intrusion is flagged if the system's behavior is later found to be inconsistent with the profile. Anomaly-based techniques typically have significant false positives or negatives, whereas signature-based detection methods, which are also referred to as misuse pattern matching, are unable to identify unknown attacks.

The distributed nature of the cloud environment makes it the environment with the greatest potential for intrusion and vulnerability to attacks. By systematically examining the configurations, network traffic, and logs, intrusion detection systems can be used to improve the security of such systems. However, conventional intrusion detection systems (IDSs), which can be broken down into host-based intrusion detection systems (HIDS) and network-based intrusion detection systems (NIDS), are not suitable for use in a cloud environment because they are unable to locate the hidden attack trail. For instance, network-based IDSs are unable to detect any event in the case of encrypted node communication, and it is possible for an attacker to take control of the installed virtual machines if the hypervisor is compromised. DKSM, Sub-Virt, and Bluepill are some of the most common attacks on virtual machines. The compromised hypervisor can be used by attackers to take control of the host. The IDS methods do not provide the same level of protection in virtualized environments because they were not developed with virtualization in mind. Because IDS cannot inspect the internal workings of operating systems, there are certain trade-offs that must be made when deploying IDS in a virtual environment. Virtualization has a lot going for

it in terms of benefits, but it also comes with a lot of security risks. It brings with it a number of brand-new issues that did not previously exist in a conventional computing environment.

Since SDN can provide a centralized system to manage the network, cloud computing providers are adopting it to achieve on-demand provisioning of network services. SDN gives the network administrator the ability to easily access and manage individual flows by making it easier for them to implement monitoring applications like firewalls and intrusion detection systems (IDSs). SDN is also an excellent choice due to the network's scalable monitoring and dynamic re-configuration requirements in the cloud.

On the basis of a set of requirements (essentially drawing from the list of requirements articulated by Patel et al.), this paper analyzes various IDS techniques proposed in the literature [11] and bolstered by a new postulate (which we propose). Patel et al.'s idea of deploying SDN in the cloud was not yet implemented conceived the list. Cloud service providers have begun to use SDN to meet their networking needs, despite the fact that SDN implementation in a cloud environment is still in its infancy. The efficient operation of IDS in a cloud environment that is based on SDN is the additional requirement that we have placed for investigation. Finally, the methods based on virtual machine introspection (VMI) have been thoroughly discussed.

There are already a few survey papers that focus on this area. However, the majority of these survey papers either cover cloud-based IDS poorly or are out of date. Zbakh and co. suggested comparing several cloud-ready IDS architectures and conducting a multi-criteria analysis. Macbeth (Measuring Attractiveness by a Categorical Based Evaluation Technique) is the only research paper to evaluate cloud-based IDS architectures using a multi-criteria decision analysis. Modi and co. and Mehmood and others have looked at various cloud computing intrusions that affect the cloud environment's confidentiality, integrity, and availability (CIA triad). Techniques for intrusion detection systems (IDS) and intrusion prevention systems (IPS) are examined. In addition, Modi et al. emphasize the significance of the IDS deployment position in attaining the desired level of security. Oktay and co. enlist the different types of attacks in the cloud, followed by specific IDS models that can resist them. Patel and co. and Premathilaka and others after conducting a review and highlighting how traditional IDS fails to deliver, stress designing IDS, especially for cloud environments, keeping in mind its paradigm. In addition, Patel et al. provide us with a list of requirements for analyzing a cloud-based intrusion detection or prevention system based on NIST's

descriptions of cloud computing systems; Techniques up until the middle of 2012 have been discussed and compared.

Intrusions In Cloud

An intrusion is an attempt to compromise a system or network's confidentiality, integrity, or availability. The cloud is frequently impacted by important intrusion categories that are discussed in this section. A presentation of various cloud-based attacks that are categorized according to the cloud's deployment model follows.

Denial of Service (DoS) Attack: A hacker uses bots (zombies) to flood a system with a lot of packets and make it impossible to access the resources that are available. As a result, the services are currently unavailable on the Internet. A DoS attack on the cloud can affect more users, according to some vulnerability experts.

Insider Attack: An insider is a cloud service provider employee or associate who has privileged access to the cloud environment and the authority to make changes [8]. Because they have information about the user and the provider, insider attacks are organized. This is fatal because many attacks can be carried out inside, and in the absence of appropriate controllers, an intruder can easily evade detection. An insider launched a distributed denial-of-service (DoS) attack on Amazon Elastic Compute Cloud (EC2)[2]. This attack broke cloud users' confidentiality.

User to Root (U2R) Attack: In this type of attack, the attacker gets into the credentials of a real user and then uses system vulnerabilities (buffer overflow) to get into the root user's account. In order to gain root access to a virtual machine or host in the cloud, the attacker first gains access to an instance and takes advantage of the flaws in that instance. The cloud's integrity is being compromised by this attack[3].

The attacker uses port scanning to acquire information about open, closed, filtered, and unfiltered ports[3]. After that, the attacker uses this information to attack open ports. Port scanning is carried out using a variety of methods. The cloud's integrity and confidentiality are the targets of this attack.

Attacks On Virtualization: If the hypervisor is compromised, virtual machines can be easily accessed[5]. Exploiting a zero-day vulnerability is the most effective method for capturing virtual machines using a hypervisor. Exploiting vulnerabilities for which the system administrator or developer has not applied a patch is known as a zero-day attack. Side channel data is vulnerable to this kind of access among virtual machines because many virtual machines use the same hardware resources[2].

Backdoor Channel Attack: This is a passive attack in which a cloud node is hacked and then used as a bot to launch DDoS attacks in the future. Shellcode, Trojans, and other similar exploits compromise the system. The intruder has full access to the system and available data once the node is compromised[5].

In Fig. 1, the deployment model (SaaS, PaaS, and IaaS) has been used to identify and classify intrusions.

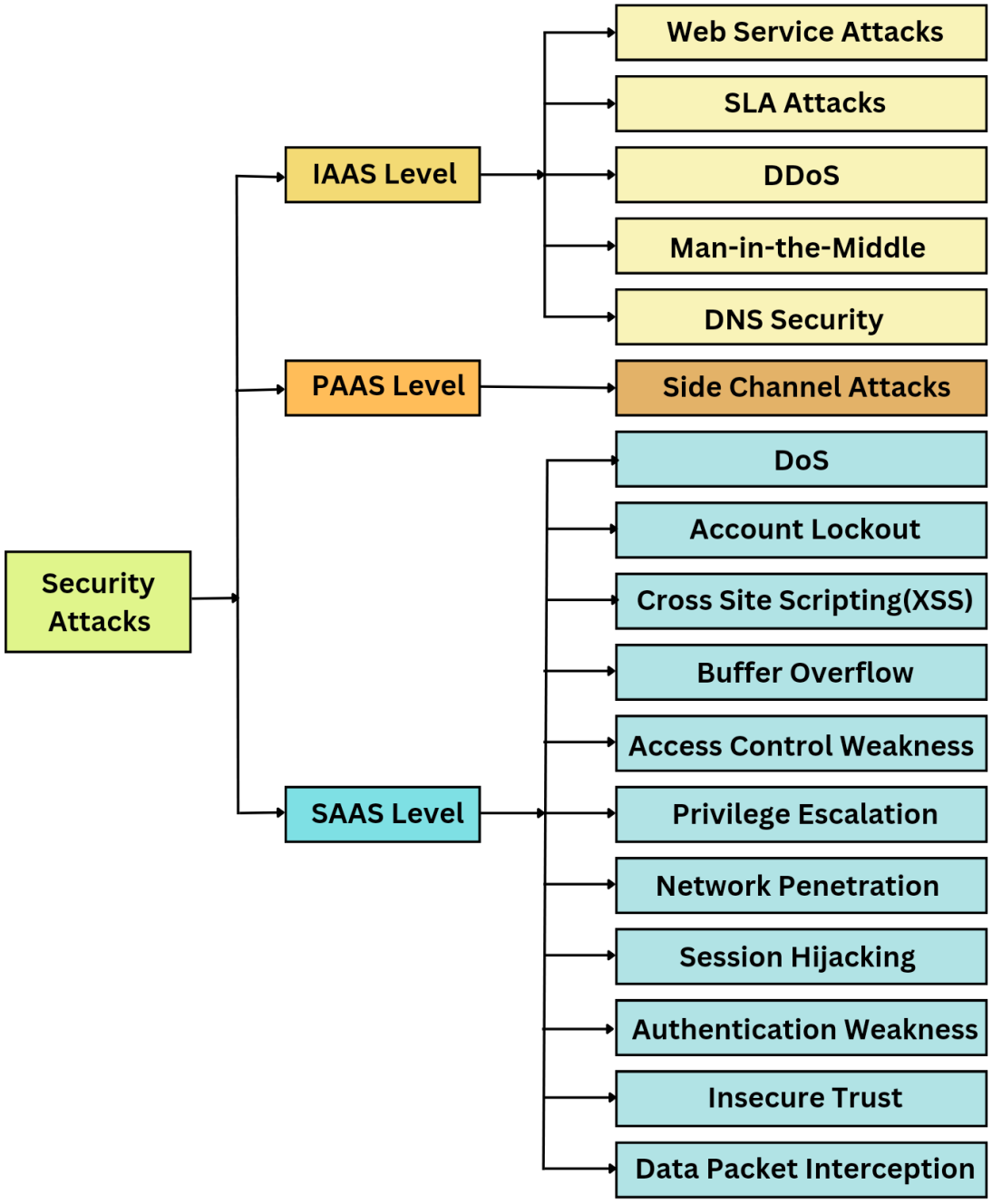

Fig. 1. Types Of Security Attacks In Cloud On The Basis Of Deployment Model.

State Of The Art Ids Architectures In Cloud

Conventional IDS, such as HIDS and NIDS, are not suitable for virtual systems, as stated in Section 1. Researchers have contributed a reasonable amount of work to the field of cloud-based IDS as a result of the emerging threats to cloud security. As depicted in Figure, the architectures discussed in this paper are broken down into four categories based on how IDS are deployed in the cloud: HIDS, NIDS, DIDS, and VMI Techniques. 2.

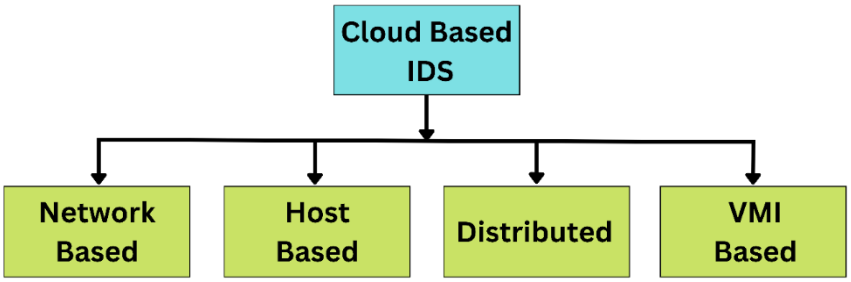

Fig. 2. Types of Cloud-Based IDS.

All of these deployment models can use anomaly detection or signature detection techniques. This section covers a survey of all these IDSs suited for the cloud.

Host Intrusion Detection Systems

There are three primary deployment-based classifications that can be applied to HIDS techniques in relation to the cloud as a whole. HIDS can be installed in the host OS to monitor the VM (where it can communicate with the guest OS to monitor either the host OS or the guest OS) or in a separate guest OS.

Low attack resistance is one disadvantage of the first scenario, in which the customer would have complete control over the IDS. It has been marked as unsuitable for the virtual cloud because it has been overwhelmingly rejected in the literature. Laureano and co A type I environment is one in which the VMM is the only process running on the host and multiple VMs run over it; a type II environment is the other. On the other hand, a type II environment implies that the VMM runs as the host machine's software. On the host machine, both regular host processes and VMM, which controls the VM, run.

Patel and co. utilize the principles of automatic computing to propose an autonomous agent-based intrusion prevention. In order to identify malicious events, an anomaly-based detection method employs autonomous sensors to monitor system activities and network traffic. Lee and co. suggest a method for identifying suspicious intrusion behavior based on the user's unusual use of resources. The authentication, authorization, and accounting (AAA) component is the technique's primary module. The database stores the user's most recent

usage history as the basis for the anomaly level. Because IDS with medium and low security use fewer resources, more guest operating systems can be added without worrying about detection speed. The administrator can perform audits on the log files.

Vieira and co. propose (GCCIDS) an intrusion detection system for cloud and grid computing. At the middleware layer, it combines techniques based on knowledge and behavior to detect intrusions. Each node is able to detect intrusion and send alerts to other nodes in a cooperative manner. Dhage and co. Before the user can access cloud services, it is suggested that an IDS controller install an IDS instance between the cloud service provider (CSP) and the user. The log files from all of the running IDS instances are gathered by the IDS controller. The controller is aided in maintaining a knowledge base for each user based on their activities by the log files. Additionally, it assists IDS in identifying the user the next time they log in. A workaround is suggested because knowledge-based IDS can only update new samples using the neural network.

Network Intrusion Detection Systems

Bakshi and co. propose a standard NIDS for virtualized environments for DDoS attack detection. On a virtual switch, where all VM traffic flows together, is installed a NIDS. In a conventional computing environment, this is comparable to the installation of a NIDS at the boundary server. The method is very similar to a standard NIDS. However, the authors have modified and tested it specifically for the virtualized environment.

All VMs' inbound and outbound traffic is logged by the NIDS on the vSwitch. DoS and DDoS attacks are detected by means of the SNORT[3] tools. The originating IP address is the basis for traffic analysis. The targeted application is transferred to a different data center and any IP address that is found to be sending a significant amount of abnormal traffic is blocked. Both complete botnets and DDoS attacks can be detected using this method. However, the paper does not discuss any performance results, and since SNORT is used for detection, only known attacks can be detected[10]. This paper does not discuss support for large virtual networks with a lot of traffic[6]. The large virtual network will make it difficult for NIDS to process all of the packets, making it possible that it will not catch attacks in time.

Mazzariello and others have deployed simulated IDS at various cloud locations to identify

DoS attacks on virtual SIP-based hosts

It is a method of detection based on signatures. SNORT has been chosen as the network IDS, and the Eucalyptus cloud computing environment has been used for testing. Gupta and co. discuss the main drawbacks of previous methods and propose a solution that addresses those

issues. The process of checking every VM for all attacks adds complexity. Profile-based IDS, they claim in their paper, will alleviate this issue. They suggest a cloud-based NIDS-based VMI that creates a unique profile for each virtual machine based on comparison with known attack signatures and deviation from normal thresholds.

Performance Analysis Of Existing Cloud Based IDs

In the preceding section, various approaches to cloud intrusion detection have been discussed, each addressing distinct research gaps. However, each method also has its own advantages and disadvantages. Low attack resistance is a flaw that comes with using a HIDS on a virtual machine. It provides excellent system visibility, but a HIDS on the host system does not support virtual machines. It would only perform the usual function of monitoring the host system itself as a HIDS. When used in conjunction with other methods, such as VMI-based, it may be effective in protecting the host system itself.

Each virtual machine can have a NIDS installed. However, deploying a NIDS on a virtual switch is the most common method. Because one component must handle all traffic and separate it, this method requires a lot of computational overhead. Due to this complexity, the IDS may fail in environments with a lot of traffic, rendering all detection results unreliable. There may even be a severe DoS attack until the NIDS is restarted if the only route for all traffic is suspended for this reason[29]. Additionally, a NIDS would not be able to detect attacks that take place within the hypervisor because it has limited visibility into the VM. Any encrypted traffic cannot be analyzed by a NIDS either. A NIDS, on the other hand, is highly resistant to attacks.

Techniques based on VMI or VMM assume that the hypervisor remains safe and free of malicious code. However, given that VMM's code is small and therefore less likely to contain bugs, this is a widely held assumption. Modi and co. mentioning that the Trusted Cloud Base typically includes a vSwitch and hypervisor. The following are the reasons they give:

1. There are fewer bugs and smaller lines of code.
2. Utilizing a Trusted Platform Module (TPM) can further enhance their security.
3. The cloud service provider has full control over them.

Wang and co. However, they argue that this assumption is not always valid because many VMMs do have a large code base. According to them, the National Vulnerability Database (NVD) discovered 18 vulnerabilities in VMware and 26 vulnerabilities in Xen Hypervisor between 2007 and 2010.

VMI techniques provide better system visibility and address the issue of attacker manipulation when compared to NIDS. Even if the guest OS has been compromised, a hypervisor-level IDS will continue to function reliably as long as the kernel data structures remain intact. The fact that the VMI remains hidden from the attacker is another factor that influences the effectiveness of VMI/VMM-based techniques. Gar-Finkel and Others mention that due to the distinct execution of instructions and operations, a VMM's presence cannot be concealed at all. In general, this has a negative impact on VMI techniques' performance.

The issue of semantic gap has also been the subject of research[10]. A VMI method collects the VM's hardware level view outside of the guest OS. As a result, visibility is reduced in comparison to that of in-guest IDS. These methods attempt to construct a high-level view of the system by utilizing knowledge of OS algorithms and kernel data structures. However, this requires a delay in detection.

Real-time monitoring is provided by other solutions when some monitoring code is installed within the guest OS. The code that monitors the virtual machine makes it easier to get a high-level view. However, there are some drawbacks to using this method. The deployed code's behavior can be altered by the presence of monitoring code[10]. In addition, if it is installed within the guest OS, it could indicate that the code is turned off before the guest OS has completely shut down and becomes operational only after it is booted[10]. Additionally, the monitoring code adds to the guest OS's computational burden. When this method is used, the guest OS must also be suspended for analysis.

Ibrahim et al.'s implementation of CloudSec claim that their method bridges the semantic gap while providing real-time monitoring. However, this method still has some instruction suspension. In order to maintain real-time monitoring, VM operations are halted whenever a memory page is sent to the Semantic Gap Bridge (SGB) for analysis by the back-end module.

The nature of VMI techniques is extremely complex. It is necessary to have current and precise information about the internal workings of the particular operating system. Even though this is a difficult procedure for an open-source system, reverse engineering would be necessary for closed-source systems as well[11]. In addition, any installed patch or system update would result in changes to all data, rendering the introspection tool useless.

Dolan-Gavitt and others A three-phase method for automatically creating an introspection tool is described in[51]. A code snippet embedded within the operating system stores all OS-related information during the initial training phase. Information that is related to security or that is specifically required for introspection is extracted during the second phase, analysis.

The data is then used to create an introspection program that can run outside of the guest OS, specifically in phase 3's runtime environment. The technique described in is constrained by the timing issue that arises when reconstructing semantic views. This is because the IDS detects attacks using a comparison-based method. If guest OS and VMM views are not exactly synchronized, there will always be some differences. This may result in numerous false negatives. Because all VMI-based techniques rely on the assumption that VMM is from the trusted code base, it is mentioned as a limitation when VMM code is modified. Any alterations to its code may result in significant flaws, opening the door to potential attacks.

Payne and co. argue that the trampoline's hook and functionality are security bottlenecks. It is evident that the code still resides inside the monitored VM and is susceptible to attacks, despite the authors' claim that the code for these functions is very small and contained, making it reliable. For the purpose of detecting attacks, the authors use logging functionality in in. The tool can be used inside or outside of the virtual machine. It would not withstand attacks if it were installed within the VM. A limitation is the timing overhead that occurs when recording, replaying, and analyzing the attacks. Before the IDS can detect an attack, the malicious entity may cause significant damage.

Virtualization is mentioned as a method for dealing with malicious mobile agents in the DIDS technique. Mobile agents, on the other hand, are installed within a monitored VM for the virtual cloud. It has not been discussed how shifting their position to another VM would affect performance. Due to the potential for significant overhead, it has been cited as a limitation.

Patel and co. mention eight performance requirements for any cloud-based IDS techniques. Six of these requirements have been utilized as an additional method for evaluating the methods discussed in this paper.

Common Challenges And Open Issues Of IDs

In order to improve security and protection, intrusion detection methods have improved with network and computer infrastructure development. There are still unsolved issues in this field, despite extensive research. The lack of standard metrics and assessment methods, low throughput and high cost of intrusion detection systems, and encrypted data are among the most significant obstacles to their implementation.

Low detection efficiency is the result of a high rate of false positives. In anomaly-based IDS, less time spent training results in more false positives, while more time spent training makes use of more resources. It is necessary to strike a balance between security and usability.

Wideband technologies' high data rates (Gbps) result in low throughput and high IDS costs. Grid computing-based and distributed detection methods are proposed as a solution to this issue. Due to a lack of standard metrics and assessment methods, selecting IDS is difficult. Axelsson et al. assert that According to the report, IDS themselves are attacked, but no security measures are taken to protect them. Ptacek and co. propose various security mechanisms for the IDS[8]. Encrypted data is one of the most significant challenges that IDS must overcome across all platforms. An IDS's design and implementation should take into account the aforementioned considerations.

Lack Of Datasets For Cloud IDS

The attacks on cloud computing have evolved over time, and it has been observed that the lack of datasets makes it difficult to set up an effective intrusion detection system. Due to the diversity of user requirements, cloud data size, and heterogeneous operating systems installed in virtual machines, traditional computing datasets cannot be utilized.

Kholidy and co propose the only dataset specifically designed with cloud infrastructure in mind, a cloud intrusion detection dataset (CIDD). It is made up of user-based audit and knowledge data that was gathered from Windows and Unix users. Audit parameters included in CIDD can identify host-based, network-based, and masquerade attacks. However, there is still insufficient data in the dataset to enable a more extensive detection.

The true and false positive rates of an intrusion detection dataset determine its effectiveness. Sending attacks to the cloud IDS and analyzing the number of attacks that are found is how the true positive rate is calculated. The ratio of false alerts is determined by the false positive rate. An ideal IDS will have zero false positives, or false alerts, generated. There are a number of major reasons why creating a cloud dataset is difficult, which are outlined below:

1.It is not possible to research models and solutions using data from actual attacks. Researchers cannot examine the data that is labeled as evidence following an attack.

2.A researcher developing an attack scenario will have a difficult time controlling the infrastructure of a commercial cloud. Additionally, the similarities between private cloud vulnerabilities and conventional IT infrastructure hinder the development of attack scenarios.

3.It is challenging to collect data from various users due to the variety of operating systems in VMs, such as Unix and Windows.

4.The large amount of audit data and the large number of cloud computing users necessitate powerful computing resources.

How To Detect Application Level DDoS Attacks In Cloud

One of the most dangerous types of distributed denial of service (DDoS) attacks is the application level DDoS flooding attack because it uses less bandwidth and is more subtle than a volumetric attack. The application-level DDoS flooding attack has the same effect on services as the volumetric attack because it targets specific application features like HTTP and DNS.

Application-level DDoS flooding attacks are responsible for a rising number of incidents, according to Gartner research[11]. Additionally, access to the payload's information is required for their mitigation. While cloud-based IDS can currently detect application layer attacks, they are unable to detect DDoS attacks that make use of valid packets. Today, the majority of attackers use valid packets. Anomaly-based intrusion detection systems (IDS) can, to a certain extent, detect these types of attacks, but experts must manually tune them, and IDS may not be able to detect all attack flows. In addition, IDS only detects threats and issues an alarm. There are many problems with using IDS as a DDoS defense platform because they won't do anything to stop the attack. IDS cannot function without a complementary mitigation strategy that can identify extremely complex attack flows and take the necessary measures. Application layer DDoS attacks will be missed by signature-based IDS. IDS is not optimized enough to detect and mitigate DDoS, so significant efforts are required to propose a solution that is a perfect trade-off between performance and security. Sophisticated DDoS attacks are identified by anomalous behavior at L3 and L4.

Conclusion

IDS ensure the confidentiality, integrity, and availability of a computer system. IDSs for cloud computing are in high demand as the number of cloud users grows at an exponential rate. We have discussed existing cloud-based intrusion detection solutions in this paper. There are four categories of cloud-based IDS, which include: systems that are network-based, host-based, distribution-based, and based on virtual machine introspection The performance that is required for a cloud-based IDS in general is mapped to their unique capabilities as well as their limitations. The basic characteristics of a cloud serve as the basis for the performance criteria used in the evaluation, and an additional requirement has been proposed, which is the efficient operation of an intrusion detection method in an SDN-based cloud environment. Software-based traffic analysis, a global network view, and centralized control are all features of SDN. SDN, on the other hand, has its own set of security issues.

In addition, common pitfalls associated with IDS implementation have been discussed. The major issues, such as the lack of datasets to evaluate an IDS's performance in the cloud, the

method for detecting application-level DDoS attacks in the cloud, the security of SDN, the secure hypervisor, VM migration, and an effective and efficient IDS in a cloud architecture, have been brought to light. In conclusion, a great deal of work has been done to identify cloud using IDS. However, the evolving nature of cloud, including scalability, distributed processing, big data analysis, and service-oriented architecture, means that the ideal IDS still needs to be improved.

References

[1]. Negi, P. S., Garg, A., & Lal, R. (2020, January). Intrusion detection and prevention using honeypot network for cloud security. In 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence) (pp. 129-132).

[2]. Shyla, S. I., & Sujatha, S. S. (2020). Cloud security: LKM and optimal fuzzy system for intrusion detection in cloud environment. Journal of Intelligent Systems, 29(1), 1626-1642.

[3]. Abdulqadder, I. H., Zhou, S., Zou, D., Aziz, I. T., & Akber, S. M. A. (2020). Multi-layered intrusion detection and prevention in the SDN/NFV enabled cloud of 5G networks using AI-based defense mechanisms. Computer Networks, 179, 107364.

[4]. Shamshirband, S., Fathi, M., Chronopoulos, A. T., Montieri, A., Palumbo, F., & Pescapè, A. (2020). Computational intelligence intrusion detection techniques in mobile cloud computing environments: Review, taxonomy, and open research issues. Journal of Information Security and Applications, 55, 102582.

[5]. Alkadi, O., Moustafa, N., Turnbull, B., & Choo, K. K. R. (2020). A deep blockchain framework-enabled collaborative intrusion detection for protecting IoT and cloud networks. IEEE Internet of Things Journal, 8(12), 9463-9472.

[6]. Guezzaz, A., Asimi, A., Asimi, Y., Azrour, M., & Benkirane, S. (2021). A distributed intrusion detection approach based on machine leaning techniques for a cloud security. In Intelligent Systems in Big Data, Semantic Web and Machine Learning (pp. 85-94). Springer, Cham.

[7]. Mondal, A., & Goswami, R. T. (2021). Enhanced Honeypot cryptographic scheme and privacy preservation for an effective prediction in cloud security. Microprocessors and Microsystems, 81, 103719.

[8]. Rani, S. (2021). A Perspective for Intrusion Detection & Prevention in Cloud Environment. International Journal of Advanced Networking and Applications, 12(6), 4770-4775.

[9]. Kumar, A., Umurzoqovich, R. S., Duong, N. D., Kanani, P., Kuppusamy, A., Praneesh, M., & Hieu, M. N. (2022). An intrusion identification and prevention for cloud computing: From the perspective of deep learning. Optik, 270, 170044.

[10]. Lata, S., & Singh, D. (2022). Intrusion detection system in cloud environment: Literature survey & future research directions. International Journal of Information Management Data Insights, 2(2), 100134.

[11]. Alghamdi, S. A. (2022). Novel trust-aware intrusion detection and prevention system for 5G MANET–Cloud. International Journal of Information Security, 21(3), 469-488.

[12]. Chen, J., Zhang, X., Wang, T., Zhang, Y., Chen, T., Chen, J., ... & Liu, Q. (2022, June). Fidas: Fortifying the cloud via Comprehensive FPGA-based Offloading for intrusion detection: industrial Product. In Proceedings of the 49th Annual International Symposium on Computer Architecture (pp. 1029-1041).

[13]. Onyema, E. M., Dalal, S., Romero, C. A. T., Seth, B., Young, P., & Wajid, M. A. (2022). Design of intrusion detection system based on cyborg intelligence for security of cloud network traffic of smart cities. Journal of Cloud Computing, 11(1), 1-20.